

## یونٹ نمبر 1 پروگرامنگ کا تعارف



C لینگوچ ڈنیس رچی نے 1969  
اور 1973 کے درمیان نیل  
لیبارٹریز میں تیار کی۔

س: کپیوٹر پروگرام کیا ہے؟ کپیوٹر پروگرامنگ کے تصور کی وضاحت کریں۔  
کپیوٹر کو دی گئی بدایات کے سیٹ کو کپیوٹر پروگرام یا سافت ویئر کہا جاتا ہے اور ان بدایات کو کپیوٹر میں لکھتے یا ذخیرہ کرنے کے عمل کو کپیوٹر پروگرامنگ کہا جاتا ہے۔ جو شخص کپیوٹر پروگرام کو صحیح طور پر لکھنا جانتا ہے اسے پروگرامر کے طور پر جانا جاتا ہے۔ کپیوٹر اگریزی، اردو یا کسی اور عام لینگوچ کو نہیں سمجھ سکتا ہے انسان ایک دوسرے کے ساتھ بات چیت کے لئے استعمال کرتے ہیں۔ ان کی الہنی خاص لینگوچ ہیں، جو کپیوٹر پروگرامرز نے ذیڑ ان کی ہیں۔ پروگرامر ان خصوصی لینگوچ میں کپیوٹر پروگرام لکھتے ہیں جنہیں پروگرامنگ لینگوچ کہا جاتا ہے۔ C, C++, C#, JAVA اور Python سب سے زیادہ استعمال ہونے والی پروگرامنگ لینگوچ ہیں۔

س: ہمیں پروگرامنگ انوائرنمنٹ کی ضرورت کیوں ہے؟

کسی بھی کام کو صحیح طریقے سے انجام دینے کے لیے ہمارے پاس مناسب نولز (Tools) ہوتا ہے۔ مثال کے طور پر باغبانی کے لیے ہمیں باغبانی کے آلات درکار ہیں اور پینٹنگ کے لیے ہمیں پینٹ، برش اور کیوس کی ضرورت ہوتی ہے۔ اسی طرح ہمیں پروگرامنگ کے لیے مناسب نولز کی ضرورت ہے۔ پروگرامر کو پروگرامنگ کے لیے مناسب نولز کی ضرورت ہوتی ہے۔ پروگرامنگ کے تمام ضروری نولز کا مجموعہ پروگرامنگ کا انوائرنمنٹ ہوتا ہے۔ پروگرام لکھنے سے پہلے پروگرامنگ انوائرنمنٹ قائم کرنا ضروری ہے۔ یہ ہمارے لیے پروگرام لکھنے اور اس پر عمل کرنے کے لیے ایک بنیادی پلیٹ فارم کے طور پر کام کرتا ہے۔

س: انگریزی ڈالپیٹ انوائرنمنٹ (IDE) کیا ہے؟ سی پروگرامنگ لینگوچ کے لیے آئی ڈی ای لکھیں۔

سافت ویئر جو کپیوٹر پروگرام لکھنے اور چلانے میں پروگرامرز کو سہولت دینے کے لیے پروگرامنگ کا انوائرنمنٹ فراہم کرتا ہے اسے انگریزی ڈالپیٹ انوائرنمنٹ (IDE) کہا جاتا ہے۔ IDE کا ایک گرافیکل یوزر انٹرفیس (graphical User Interface) ہوتا ہے۔ اس کا مطلب یہ ہے کہ کوئی صارف ان پٹ فراہم کرنے اور آؤٹ پٹ حاصل کرنے کے لیے وندوز اور بننوں کا استعمال کرتے ہوئے اس کے ساتھ بات چیت کر سکتا ہے۔ ایک IDE ایسے نولز پر مشتمل ہوتا ہے جو کپیوٹر پروگرام لکھنے، چلانے اور جاپنے کے تمام مرحلے میں ایک پروگرامر کی مدد کرتا ہے۔ یہ ایک ہی انٹرفیس میں یکجنت ایڈیٹر، کپیلر اور ڈیبگر کو ملا کر حاصل کیا جاتا ہے۔ C پروگرامنگ لینگوچ کے لیے کچھ IDEs یہ ہیں:

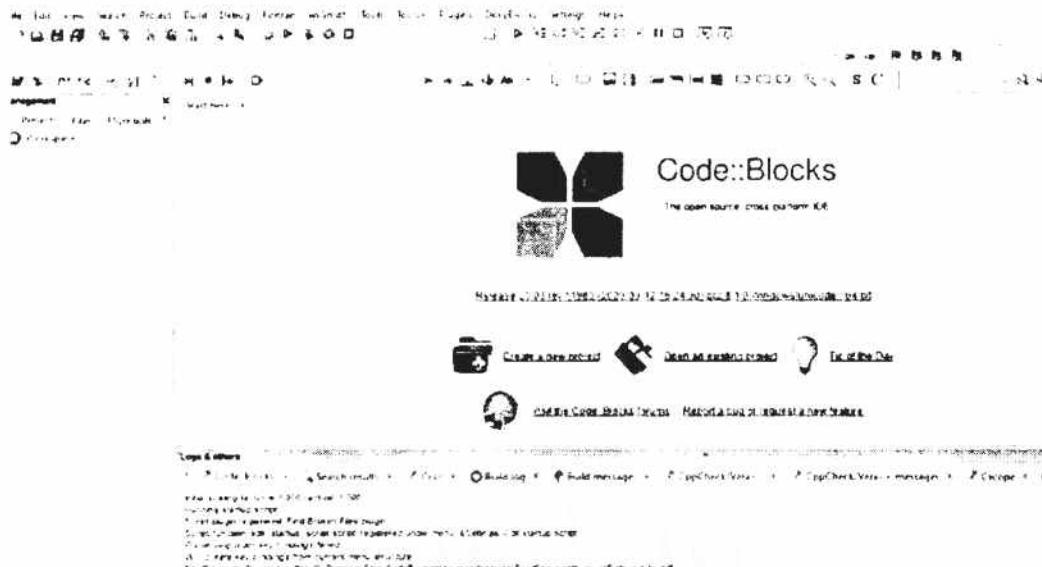
Dev C++ (4)

Code :: Blocks (3)

X Code (2)

Visual studio (1)

main کی Code :: Blocks میں سکرین ہے۔



main کی Code :: Blocks ایٹریس

IDE مندرجہ ذیل دو حصوں پر مشتمل ہے۔

1. نیکست ایڈیٹر (Text Editor)

نیکست ایڈیٹر زایک سافت ویربے جو پروگرامز کو کمپیوٹر پر گرام لکھتے اور ایڈٹ کرنے کی اجازت دیتا ہے۔ تمام IDEs کے اپنے مخصوص

نیکست ایڈیٹر ہوتے ہیں۔ ہم IDE کی میں سکرین پر اپنے پروگرام لکھ سکتے ہیں۔



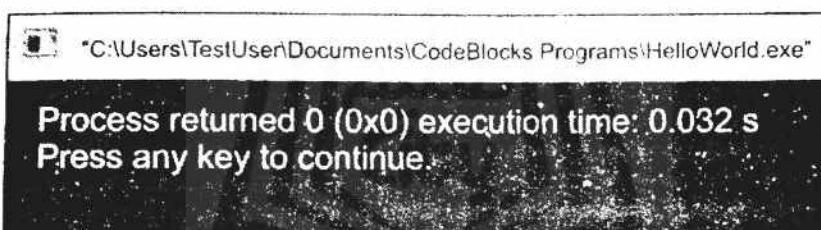
نیکست ایڈیٹر کا Code :: Blocks

IDE Code :: Blocks کے نیکست ایڈیٹر میں لکھا گیا بنیادی C لینگوچ کا پروگرام ہے۔ جب اس پروگرام پر عملدرآمد ہوتا ہے تو یہ "Hello World" دکھاتا ہے۔ کمپیوٹر سکرین پر ہمیں اپنی فائل کو چلانے سے پہلے تحفظ کرنا ہو گا۔ ہم نے اپنے پروگرام فائل کو "Hello World.c" کا نام دیا ہے ہم پروگرام کی آٹھ پت دیکھنے کے لیے "build and run" ہن پر کلک کر سکتے ہیں۔

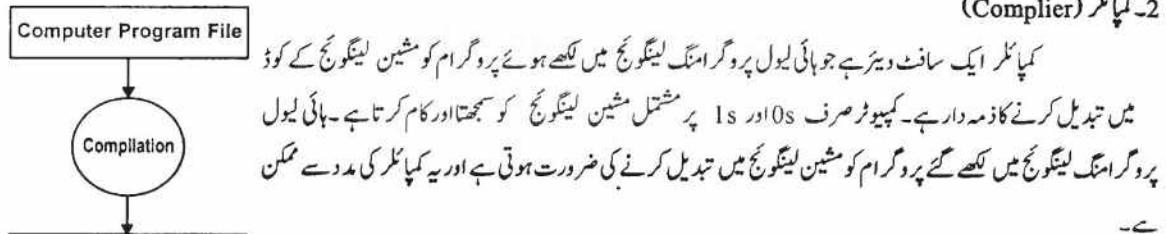


Code :: Blocks میں ایکسپلور ہوتا ہو اپروگرام

کنسول (Console) آٹھ پت سکرین اس طرح نظر آتی ہے۔



## 2- کپاٹر (Complier)



س: پروگرامنگ کے بنیادی نکات کی وضاحت کریں؟

### (i) لینگوچ کا سنتیکس (Syntax of Language)

ہر پروگرامنگ لینگوچ میں کچھ ابتدائی تغیراتی عناصر ہوتے ہیں اور ایک درست پروگرام لکھنے کے لیے کچھ اصول فراہم کیے جاتے ہیں۔ تواعد کا یہ مجموعہ لینگوچ کے سنتیکس کے طور پر جانا جاتا ہے۔ سنتیکس کو ایک پروگرامنگ لینگوچ کا گرامر سمجھا جاسکتا ہے۔

### (ii) سنتیکس ایر (Syntax Error)

پروگرامنگ کے دوران اگر مناسب سنتیکس یا پروگرامنگ لینگوچ کے تواعد پر عمل نہیں کیا جاتا ہے تو پروگرام چلتا نہیں ہے۔ اس صورت میں کپاٹر ایک ایر دیتا ہے۔ اس قسم کی غلطیوں کو سنتیکس ایر کہا جاتا ہے۔

## پروگرامنگ کا تعارف

06

(iii) **مختصر کے کئے الفاظ (Keywords)** کچھ مکمل کو معلوم ہوتے ہیں۔ ان الفاظ کو مخصوص الفاظ یا کی۔ ورز کے نام سے جانا جاتا ہے۔ اگر کوئی پروگرام انہیں اپنی تعریف دیتا ہے تو یہ سنیکس ایر کا جب ہتا ہے۔ درج ذیل مخصوص الفاظ سی پروگرامنگ لینگوچ میں استعمال ہوتے ہیں۔

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

سی لینگوچ میں پروگرام کی ساخت کی وضاحت کریں؟  
سی لینگوچ میں لکھے گئے پروگرام کوئی اہم حصوں میں تقسیم کیا جاسکتا ہے۔

### 1. لینک سیشن یا ہڈر سیشن (Link Section or Header Section)

سی لینگوچ میں پروگرام لکھتے ہوئے ہم ان فنکشنز کا سچ استعمال کرتے ہیں جو پہلے ہی لینگوچ میں بیان کیے گئے ہیں۔ لیکن موجودہ فنکشنز کو استعمال کرنے سے پہلے ہمیں ان کو شامل کرنے کی ضرورت ہے جہاں ان فنکشنز کی وضاحت کی گئی ہو۔ ان فائلوں کو ہڈر فائل کہا جاتا ہے۔ ہم ان ہڈر فائلوں کو پروگرام کے اپری حصے میں include statement لکھ کر اپنے پروگرام میں شامل کرتے ہیں۔ include statement کی ساخت مندرجہ ذیل ہے:

`#include<header_file_name>`

یہاں ہڈر فائل کا نام درج بالا مثال میں کسی بھی ہڈر فائل کا نام ہو سکتا ہے۔ ہم نے `<stdio.h>` شامل کی ہے جس میں ان پت اور آٹھ پٹ فنکشنز سے متعلق معلومات ہیں۔ بہت سی دوسری ہڈر فائلیں بھی دستیاب ہیں مثلاً کے طور پر `<math.h>` میں تمام پہلے سے مطہرہ ریاضی کے فنکشنز شامل ہیں۔

2. main سیشن: سی پروگرام میں ایک (main) فنکشن ہوتا ہے اور یہاں سے پروگرام چنان شروع ہوتا ہے۔

### 3. main() کی باڑی

main() کی باڑی میزگی بریکنوں ({} ) کے اندر ہوتی ہے۔ تمام سینیٹس جو میزگی بریکنوں ({} ) کے اندر موجود ہوئی ہیں وہ main() فنکشن کی باڑی میں شامل ہوتی ہیں۔ پروگرام میں printf("Hello World!"); کپیوڑا سکرین پر "Hello World" کو ظاہر کرنے کے لیے پہلے سے بیان شدہ printf استعمال کرتا ہے۔

ی لینگوچ میں پروگرام لکھنے کے اصول:

- ی لینگوچ پروگرام میں سینٹیٹس کی ترتیب اس ترتیب کے مطابق ہونی چاہیے جس ترتیب سے ہم سینٹیٹس کو چلا جاتے ہیں۔
- ی لینگوچ کیس سینٹیٹس (Case Sensitive) ہے۔ اس کا مطلب یہ ہے کہ اگر کسی ذخیرہ لفظ کو تمام چھوٹے حروف کے ساتھ بیان کیا جاتا ہے تو اسے بڑے حروف میں نہیں لکھا جاسکتا ہے لیکن int اور Int مختلف ہیں۔ جبکہ پہلا ذخیرہ الفاظ ہے اور دوسرا نہیں ہے۔
- ہر سینٹ (Statement) کے آخر پر سیکی کولن (;) آتا ہے۔

س: یہ پروگرام میں کیمیٹس کا مقصد اور سنٹیکس بیان کریں۔

کیمیٹس ایک پروگرام میں وہ سینٹیٹس ہیں جن کو کمپلائلر نظر انداز کرتا ہے اور ان پر عملدرآمد نہیں ہوتا ہے۔ عام طور پر کیمیٹس کوڈ کی تفصیل فراہم کرنے کے لیے نظری لینگوچ میں لکھتے جاتے ہیں جیسے انگریزی لینگوچ میں۔

### کیمیٹس لکھنے کا مقصد

کیمیٹس کو پروگرام کی دستاویزات کے طور پر سمجھا جاسکتا ہے۔ ان کا مقصد یہ ہے:

- 1) یہ دوسرے پروگرامز کے کوڈ کو سمجھنے میں سہولت فراہم کرتا ہے۔
- 2) یہ ہمیں اپنے کوڈ کو لکھنے کے برسوں بعد بھی سمجھنے میں مدد کرتا ہے۔ ہم نہیں چاہتے کہ ان سینٹیٹس پر عملدرآمد ہو کیونکہ اس سے سنٹیکس کی غلطی ہو سکتی ہے کیونکہ سینٹیٹس نظری لینگوچ میں لکھی ہوتی ہیں۔

### کیمیٹس کی اقسام:

ی پروگرامنگ لینگوچ میں کیمیٹس کی دو اقسام ہیں:

#### 1. سنگل لائئن کیمیٹس (Single Line Comments)

سنگل لائئن کیمیٹس // سے شروع ہوتے ہیں۔ // کے بعد کچھ بھی ایک ہی لائن پر لکھا ہو وہ کیمیٹ ہی سمجھا جاتا ہے۔ مثال کے طور پر

// This is a comment.

#### 2. ملٹی لائئن کیمیٹس (Multi Line Comments)

ملٹی لائئن کیمیٹس /\* سے شروع ہوتے ہیں اور \*/ پر ختم ہوتے ہیں۔ /\* اور \*/ کے درمیان کسی بھی چیز کو ایک کیمیٹ سمجھا جاتا ہے چاہے ایک سے زیادہ سطروں پر ہوں۔ مثال کے طور پر:

/\* This is a Multi line

Comment \*/

درج ذیل کوڈ میں کمینش کا استعمال دکھایا گیا ہے۔

### **Example**

```
#include < stdio.h >
/*this program displays "I am a student of class 10th" on the output screen*/
void main( )
{ //body of main function starts from here
printf ("I am a student of class 10th");
} //body of main function ends here
```

س: مستقلات اور اس کی اقسام کی وضاحت کرس۔

ہر لیگنگ کے حروف کا ایک بنیادی سیٹ (کریکٹر سیٹ) ہوتا ہے جنہیں قابلِ اطلاق طریقے سے جوڑ کر الفاظ بنائے جاتے ہیں اور پھر ان الفاظ کو جملے بنانے کے لیے استعمال کیا جاسکتا ہے۔ اسی طرح سی پروگرام میں لیگنگ کیس بھی کریکٹر سیٹ ہوتے ہیں۔ جس میں یہ سب شامل ہوتے ہیں:

- (A, B, ..., Y, Z), (a, b, ..., y, z)  $\models_{\mathcal{F}} \varphi$  (1-9)  $\vdash_{\mathcal{F}}$  (2)

(3) خصوصی علامتیں ( ) , “ ” , <>, . , ? / , ! \ , [ ] ; , : , ^ , & , % , # , @ , ! - + =

یہ حروف صحیح، ہند سے اور خاص علماتیں جب قابل اجازت انداز میں مل جائیں تو مستقلات، تغیرات اور ذخیرہ الفاظ (جسے کی ورد بھی کہا جاتا ہے) بناتے ہیں۔

### مُسْكَلَات (Constant)

مستقلات وہ قیمتیں ہیں جنہیں پروگرام کے چلنے کے دوران تبدیل نہیں کیا جا سکتا مثلاً 1500, 75.7، 5، وغیرہ۔ ہی لینکوگن کمپنی میں مستقلات کی قسم اقسام ہیں:

کریکٹ مسئلہ

حقیقی مسئلقات

اٹیجھ مسکات

## اُسٹریج مسٹکل اسٹنڈارڈ (Integer Constants)

ان قیتوں میں اعشار یہ نہیں آتا ہے۔ یہ ثبت یا منفی ہو سکتی ہیں۔ مثلاً 7,1256,-54,55555,30100 وغیرہ۔ اگر قیمت سے پہلے کوئی علامت نہ ہو تو اسے ثبت سمجھا جاتا ہے۔

### حثیت مسناوات (Real Constants)

ان قیوں میں اعشار آتا ہے۔ مثلاً 7941.2345,-1575.76,75.0,15.3333,14.3 - غیر ویرشت پامنی بھی ہو سکتی ہیں۔

## کریکٹر مسٹریل ات (Character Constants)

اثتیجہ اور کریکٹر مستقلات کے درمیان فرق:

ایک ہندسے بطور کریکٹر یعنی '9'، ایک ہندسے 9 سے مختلف ہوتا ہے جو کہ ایک اثتیجہ کریکٹر کے طور پر استعمال ہوتا ہے۔ ہم ریاضی کا واضح نتیجہ حاصل کرنے کے لیے دو اثتیجہ شامل کر سکتے ہیں جیسے  $17 = 8 + 9$  لیکن ہم کوئی کریکٹر مستقلات کو تبع نہیں کر سکتے۔ مثلا  $17 \neq 8 + 9$

س: متغیر کا ہے؟ متغیر کے ذیل ناتاپ کی وضاحت کریں۔

### حتمیات (Variable)

متغیر دراصل میموری لوکیشن (Memory Location) کو ریاجانے والا نام ہے، کیونکہ ذیل ناتاپ کی میموری کے اندر محفوظ ہوتا ہے۔ ایک پروگرام میں متغیر کی قیمت کو تبدیل کیا جاسکتا ہے۔ اس کا مطلب یہ ہے کہ کسی پروگرام میں اگر کسی متغیر کی قیمت 5 پر مشتمل ہو تو بعد میں ہم اسے دوسری قیمت دے سکتے ہیں جو کہ قیمت 5 کو بدلتی ہے۔ درست متغیر کے ناموں کی کچھ مثالیں درج ذیل ہیں: height, average\_weight

ہر متغیر کا ایک منفرد نام ہے جسے شناخت کرتا ہے (Identifier) کہا جاتا ہے اور ایک ذیل ناتاپ ہوتی ہے۔ ذیل ناتاپ بیان کرتی ہے کہ متغیر میں کس قسم کا ذیل محفوظ کیا جاسکتا ہے۔ C لینگوچ میں ذیل کی مختلف اقسام ہیں جیسے int, float, char اور float۔ مندرجہ ذیل جدول میں لینگوچ میں ذیل کی اقسام کو ظاہر کرتا ہے۔

Type of data	Matching data type in C language	Sample value
Integer	int	123
Real	float	23.5
Character	char	'a'

### متغیر کی ذیل ناتاپ (Data type of Variable)

یہ لینگوچ میں ہر متغیر کی ایک ذیل ناتاپ ہوتی ہے۔ ذیل ناتاپ نہ صرف متغیر کے اندر دخیرہ کیے جانے والے ذیل کو بیان کرتی ہے بلکہ یہ بھی بتاتی ہے کہ کپائلر کو اس ذیل کو محفوظ کرنے کے لیے کتنے بائیس کی ضرورت ہوتی ہے۔ C لینگوچ میں استعمال ہونے والے ذیل کی اقسام درج ذیل ہیں:

#### 1) اثتیجہ (Integer)

اثتیجہ ذیل ناتاپ اثتیجہ ولیوز کو محفوظ کرنے کے لیے استعمال کی جاتی ہے۔ ایک اثتیجہ 4 بائیس میموری لیتا ہے۔ اثتیجہ ناتاپ متغیر ذیکر کرنے کے لیے کی ورڈ لفظ int استعمال کرتے ہیں۔ اثتیجہ کی دو اقسام ہیں۔

Signed int .1

Signed int میں 2,147,483,647 سے -2,147,483,647 تک تمام ثبت اور منفی دونوں قیمتیں محفوظ کی جا سکتی ہیں۔ بطور ذیفات

ناتاپ int کو Signed int سمجھا جاتا ہے۔

Unsigned int .2

ایک int unsigned صرف ثبت قیتوں کو محفوظ کر سکتا ہے اور اس کی قیمت 0 سے  $4,294,967,295$  تک ہوتی ہے۔  
ان شعبہ ذیلی خیر کرنے کے لیے کی - ورڈ unsigned int استعمال کیا جاتا ہے۔

### (2) فلوٹنگ پوائنٹ (float)

ذینا ٹاپ ایک حقیقی نمبر (فلوٹنگ پوائنٹ کے ساتھ) اعشاریہ کے بعد زیادہ سے زیادہ جھہ بند سے آسکتے ہیں۔ float ٹاپ کا متغیر ذیلی خیر کرنے کے لیے ہم کی ورڈ float استعمال کرتے ہیں۔ ایک float میوری کے 4 بائیٹس استعمال کرتا ہے۔ اس کی قیمت  $3.4 \times 10^{-38}$  سے  $3.4 \times 10^{38}$  تک ہوتی ہے۔

### (3) کریکٹر (char)

C میں کریکٹر ٹاپ متغیرات کو ذیلی خیر کرنے کے لیے ہم کی ورڈ char استعمال کرتے ہیں۔ یہ ذخیرہ کرنے کے لیے صرف ابھت میوری لیتا ہے۔ char کا متغیر صرف ایک کریکٹر کو محفوظ کر سکتا ہے۔

سوال: ہی لینگوئیج میں متغیرات کے نام لکھنے کے قواعد کیسیں؟

(Rules for Naming Variable in C Language)

ہر متغیر کا ایک منفرد نام یا شناخت لکھندا ہو نا ضروری ہے۔ متغیر کے نام کے لیے درج ذیل اصول استعمال کیے جاتے ہیں۔

1) متغیر کے نام میں صرف حروف تہجی (بڑے یا چھوٹے) ہندے اور "—" کی علامت آسکتی ہے۔

2) متغیر کا نام ایک حرف یا اندر سکور (\_) سے شروع ہونا چاہیے یہ ہندے سے شروع نہیں ہو سکتا۔

3) ایک کی - ورڈ کو متغیر کے نام کے طور پر استعمال نہیں کیا جاسکتا۔

4) متغیر کے نام کی لسانی کا کوئی سخت اصول نہیں ہے لیکن بہتر ہے کہ نام مخفف ہو۔

ہمیں ایک متغیر کو مناسب نام دینا چاہیے جو اس کا مقصد بیان کرتا ہے جیسے کسی فحص کی تحریک کو ذخیرہ کرنے کے لیے مناسب نام salary یا

wage ہو سکتا ہے۔

سوال: متغیرات اور مستقلات میں کیا فرق ہے؟

متغیرات (Variables):

متغیر دراصل میوری لوکیشن (Memory Location) کو دیا جانے والا نام ہے، کیونکہ ذینا کچھ بزرگی میوری کے اندر محفوظ ہوتا ہے۔ ایک پر ڈرام میں متغیر کی قیمت کو تبدیل کیا جاسکتا ہے۔ اس کا مطلب یہ ہے کہ کسی پر ڈرام میں اگر کسی متغیر کی قیمت 5 پر مشتمل ہو تو بعد میں ہم اسے دوسرا قیمت دے سکتے ہیں جو کہ قیمت 5 کو بدل دیتی ہے۔ درست متغیر کے ناموں کی کچھ مثالیں درج ذیل ہیں:

height , average\_weight

**متقلات (Constants)**

متقلات وہ قیمتیں ہیں جنہیں پروگرام کے چلنے کے دوران تبدیل نہیں کیا جاسکتا مثلاً 5, 75.7, 1500 وغیرہ۔ ہی لینگوچ میں متقلات کی تین اقسام ہیں:

3. کریکٹر متقلات

2. حقیقی متقلات

1. انٹجئر متقلات

سوال: ہم ایک متغیر کو ڈیکلائریشن اور انیشلائریشن کیسے کر سکتے ہیں؟

**متغیر کو ڈیکلائریشن (Variable Declaration)**

پروگرام میں استعمال کرنے سے پہلے ہمیں متغیر کو ڈیکلائری کرنے کی ضرورت ہے۔ متغیر کو ڈیکلائری کرنے میں اس کی ذینماں پ کی وضاحت اور اسے ایک درست نام دینا شامل ہے۔ متغیر کو ڈیکلائری کرنے کے لیے مندرجہ ذیل سنکلکس کی پیروی کی جائی گی:-

Data\_type variable\_name;

متغیرات کو ڈیکلائری کرنے کی چند مثالیں درج ذیل ہیں:-

unsigned int age;

float height;

int salary;

char marital\_status;

ایک سنینٹ میں ایک ہی ذینماں پ کے ایک سے زیادہ متغیرات کو بھی ڈیکلائری کیا جاسکتا ہے جیسے درج ذیل مثالوں میں دکھایا گیا ہے۔

unsigned int age, basic\_salary, gross\_salary;

int points\_scored, steps;

float height, marks;

char marital\_status, gender;

ایک متغیر کو ڈیکلائری نہیں کیا جاسکتا جب تک کہ ہم اس کی ذینماں پ کا ذکر نہ کریں۔ متغیر کو ڈیکلائری کرنے کے بعد اس کی ذینماں پ کو تبدیل نہیں کیا جاسکتا۔ متغیر کو ڈیکلائری کرنے سے پہلے ہے کہ اس کی قسم کیا ہے، اس میں کہاں سے کہاں تک قیمتیں آئتیں ہیں اور اس پر کس قسم کے اوپر یعنی انجام دیے جائے ہیں۔ مندرجہ ذیل ایک پروگرام دکھاتی ہے جس میں دو متغیرات ڈیکلائری کیے گئے ہیں:-

```
void main ()
{
    char grade;
    int value;
}
```

## پروگرامنگ کا تعارف

**متغیر انیشلائزیشن (Variable Initialization)**

پہلی مرتبہ ایک متغیر کو قیمت سے منسوب کرنا متغیر انیشلائزیشن کہلاتا ہے۔ یہ لینگوچر میں متغیر کوڈیکسیر کرتے ہوئے یا اس کے بعد انیشلائز کیا جاسکتا ہے۔ متغیر کوڈیکسیر یعنی وقت انیشلائز کرنے کے لیے عام سنتکس یہ ہے:

**data\_type variable\_name = value ;**

مندرجہ ذیل مثال ایک پروگرام دکھاتی ہے جس میں متغیرات کوڈیکسیر اور انیشلائز کیا گیا ہے۔  
مثال

```
#include<stdio.h>
void main()
{
    char grade; //Variable grade is declared
    int value = 25; /*Variable value is declared and initialized.*/
    grade = 'A'; //Variable grade is initialized
}
```

### مشق

کس 1: درست جوابات کا انتخاب کریں۔

نمبر	سوال	A	B	C	D
1	ایک سافت ویر جو پروگرام کو کمپیوٹر پر وگرام لکھنے میں مدد دیتا ہے _____ کے نام سے جانا جاتا ہے:	کپیلر	ایڈیٹر	IDE	ڈیبگر
2	ایک سافت ویر ہے جو پروگرام کی فائلز کو ایسے کوڈ میں تبدیل کر دیتا ہے جس میں سمجھ سکے اور چلا سکے۔	کپیلر	ایڈیٹر	IDE	ڈیبگر
3	ہر پروگرام لینگوچر میں کچھ ابتدائی تغیراتی عناصر ہوتے ہیں یہ گرام کے کچھ اصولوں کے پابند ہوتے ہیں جنہیں ---- کہا جاتا ہے۔	پروگرامنگ روولز	سنتکس	تغیراتی عناصر	سیماںک روولز
4	ایسے الفاظ کی ایک فہرست جو پہلے سے طے شدہ ہے اور پروگرام اپنے متغیرات کے نام کے لیے استعمال نہیں کرتا ہے _____ کے نام سے جانا جاتا ہے۔	آنورڈ	کی-ورڈ	محدود الفاظ	پہلے سے ڈیفائن کیے ہوئے الفاظ
5	سینٹکس _____ سینٹکس میں لکھی جاتی ہیں۔	ہیڈر	مین main	کمپیونس	پرنٹ

## پروگرامنگ کا تعارف

### متغیر انیشلائزیشن (Variable Initialization)

پہلی مرتبہ ایک متغیر کو قیمت سے منسوب کرنا متغیر انیشلائزیشن کہلاتا ہے۔ یہ لینگوچر میں متغیر کوڈیکسیر کرتے ہوئے یا اس کے بعد انیشلائز کیا جاسکتا ہے۔ متغیر کوڈیکسیر یعنی وقت انیشلائز کرنے کے لیے عام سنتیکس یہ ہے:

**data\_type variable\_name = value ;**

مندرجہ ذیل مثال ایک پروگرام دکھاتی ہے جس میں متغیرات کوڈیکسیر اور انیشلائز کیا گیا ہے۔  
مثال

```
#include<stdio.h>
void main()
{
    char grade; //Variable grade is declared
    int value = 25; /*Variable value is declared and initialized.*/
    grade = 'A'; //Variable grade is initialized
}
```

### مشق

کس 1: درست جوابات کا انتخاب کریں۔

نمبر	سوال	D	C	B	A
1	ایک سافت ویر جو پروگرام کو کمپیوٹر پر وگرام لکھنے میں مدد دیتا ہے _____ کے نام سے جانا جاتا ہے:	ڈیبگر	IDE	ایمیٹر	کپیلر
2	ایک سافت ویر ہے جو پروگرام کی فائلز کو ایسے کوڈ میں تبدیل کر دیتا ہے جسے مشین سمجھ سکے اور چلا سکے۔	ڈیبگر	IDE	ایمیٹر	کپیلر
3	ہر پروگرام لینگوچر میں کچھ ابتدائی تغیراتی عناصر ہوتے ہیں یہ گرامر کے کچھ اصولوں کے پابند ہوتے ہیں جنہیں ----- کہا جاتا ہے۔	سیماںک روڈر	تغیراتی عناصر	سنتیکس	پروگرامنگ روڈر
4	ایسے الفاظ کی ایک فہرست جو پہلے سے طے شدہ ہے اور پروگرام اپنے متغیرات کے نام کے لیے استعمال نہیں کرتا ہے _____ کے نام سے جانا جاتا ہے۔	پہلے سے ڈیفارکٹ کے ہونے والے الفاظ	محدود الفاظ	کی-ورڈ	آنورڈر
5	سینٹنس یکشن میں لکھی جاتی ہیں۔	پرنٹ	کیمیٹس	مین main	ہیڈر

## پروگرامنگ کا تعارف

13

وضاحتیں	کینش	اشارے	پیغامات	کو سوس (Source) کوڈ میں پروگرام کے استعمال کے ہوئے الگوریتم اور طریقہ کار کی مزید وضاحت کرنے کے لیے استعمال کی جاتا ہے۔	6
کینش	سٹرنگز	ستقلات	متغيرات	وہ کینشیں ہیں جو پروگرام کے پورے عمل کے دوران تبدیل نہیں ہوتی ہیں۔	7
6	5	4	3	ایک float میموری کی بائنس استعمال کرتا ہے۔	8
?	@	=	→	ایک متغير کو ایشلائز کرنے کے لیے ہم آپریٹر استعمال کرتے ہیں۔	9
مجموعہ	متغير	جار	باکس	کو کائنشیں محفوظ کرنے کے لیے مرتبان سمجھا جاسکتا ہے۔	10

جوابات

1	2	3	4	5	6	7	8	9	10
B	A	B	B	A	C	B	B	B	C

س2: غلط / درست کی نشاندہی کریں۔

- 1) ایک IDE نیکست ایڈیٹر، کپاٹر اور ذیگر زکو ایک ہی اثر فیں میں اکھا کرتی ہے۔
- 2) کپیوڑ کو پروگرام فائل میں لکھنے گئے کوڈ کو چلانے کے لیے باائزی لینگوچ میں تبدیل کرنے کی ضرورت ہوتی ہے۔
- 3) C پروگرام لینگوچ میں column ایک ذخیرہ الفاظ ہے۔
- 4) comment goes here ایک درست کہت ہے۔
- 5) float میں چہ ہندسوں تک پریساائز نمبر محفوظ کیا جاسکتا ہے۔

جوابات

1	2	3	4	5
درست	درست	غلط	غلط	درست

س3: درج ذیل کی تعریف کریں۔

- 1) صفحہ نمبر 03 پر ملاحظہ کریں IDE
- 2) کپاٹر
- 3) کی-ورڈ
- 4) پروگرام کا میں سکھن
- 5) char ذیلی تابع

## پروگرامنگ کا تعارف

س 4: مختصر طور پر درج ذیل سوالات کے جواب دیں۔

صفحہ نمبر 03 پر ملاحظہ کریں

1) یہیں ایک پروگرامنگ انواع نہیں کی ضرورت کیوں ہوتی ہے؟

2) اپنے کمپیوٹر لپ کے IDE میں C پروگرام فائل بنانے کے لیے اقسام کیسیں؟

جواب: کسی پروگرام بنانے کے لیے درج ذیل اقسام استعمال کیے جاتے ہیں۔

3. کنسول ایڈیشن بہن پر کلک کریں۔

2. نیا پروجیکٹ پر کلک کریں۔

6. پروجیکٹ کا نام ناچ کریں تکسٹ بہن پر کلک کریں۔

4. GO بہن پر کلک کریں۔ 5. سی لینگوچ منتخب کریں۔

8. اب آپ اپنا پروگرام ناچ کر سکتے ہیں۔

7. Finish بہن پر کلک کریں۔

صفحہ نمبر 05 پر ملاحظہ کریں

3) کپاٹر کے مقدمہ کی وضاحت کریں؟

صفحہ نمبر 06 پر ملاحظہ کریں

4) ہی پروگرامنگ لینگوچ کے پانچ کی درود کی فہرست تحریر کریں؟

صفحہ نمبر 06 پر ملاحظہ کریں

5) ہی پروگرام کی ساخت کے اہم حصے بیان کریں؟

صفحہ نمبر 07 پر ملاحظہ کریں

6) پروگرامنگ میں کیسیں کیسیں کیوں استعمال کرتے ہیں؟

صفحہ نمبر 28 پر ملاحظہ کریں

7) کائنات اور خیرات میں فرق کریں۔

صفحہ نمبر 10 پر ملاحظہ کریں

8) خیرات کے نام رکنے کے اصول تحریر کریں۔

صفحہ نمبر 09-10 پر ملاحظہ کریں

9) char اور int کے درمیان فرق کریں؟

صفحہ نمبر 11 پر ملاحظہ کریں

10) ہم ایک خیر کو کس طرح ڈیکلائر اور انسٹال اسٹر کر سکتے ہیں؟

س 5: کالم طالیں۔

C	B	A
C Lion (d)	(a) ایسا کوڈ جو مشین پر چلایا جائے	IDE (1)
NotePad (f)	(b) include نہیں	ایڈیٹر (2)
(a) ایسا کوڈ جو مشین پر چلایا جائے	Python (c)	کپاٹر (3)
Python (c)	C Lion (d)	پروگرامنگ لینگوچ (4)
struct (h)	/•(a+b)• / (e)	ذخیرہ الفاظ (5)
include (b)	Notepad (f)	نکسکشن (6)
{ } (j)	int weight (g)	Main() کی بازی (7)
/•(a+b)• / (e)	struct (h)	کینٹ (8)
int weight (g)	◇ (i)	اشیاء متغیر (9)
◇ (i)	{ } (j)	فلوچارٹ (10)

پروگرامنگ کی مشقیں

مشن نمبر 1

- استاد محترم کی مدد سے اپنی لیب کے کمپیوٹر پر پروگرام لکھنے کے لیے انسال کی ہوئی IDE کو لیں۔
- ایسٹریم میں یہ پروگرام لکھیں اور اسے welcome نام لکھ کر محفوظ کریں۔

<pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; void main() {     /*A simple C language program*/     printf ("Welcome to C language");     getch(); }</pre>	<p>آؤٹ پٹ</p> <p>Welcome to C language</p>
---	--

مشن نمبر 2: ایک پروگرام لکھیں جو آپ کے بہترین دوست کا ذیاً محفوظ کرنے کے لیے مناسب ذیاً اپنی کمپیوٹر کے متغیرات کو ذیاً بدل کرے۔ ان متغیرات کو درج ذیل ذیاً کے ساتھ انشیلا کر کریں:

- آپ کے نام کا ابتدائی حرف
- آپ کی جنس کا ابتدائی حرف
- آپ کے تاریخیں

<pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; int main() {     char n='M' , g='M';     int age=20;     float height=6.1;     printf( " %c %c %d %f ", n , g , age , height );     getch();     return 0; }</pre>	<p>آؤٹ پٹ</p> <p>M M 20 6.1</p>
---	---------------------------------

## سرگرمیاں

سرگرمی 1.1: اپنے ویب براوزر (Web Browser) کے ذریعے C پروگرامنگ کی تین مختلف IDEs کے نام معلوم کریں۔

حل:

1. Eclipse
2. NetBeans
3. Sublime Text

سرگرمی 1.2: اپنی لیب کے کسیوں میں انشال کی ہوئی IDE کھویں۔ اپنے IDE کے نیکست اینڈیمیٹر میں پروگرام لکھیں اور اسے ایگزیکیوٹ کریں۔

حل:

```
#include<stdio.h>
void main()
{
    printf("Hello World")
}
```

سرگرمی 1.3: ان الفاظ میں سے C لینکنگ کے ریزروڈ یا مخفی کے گئے الفاظ کے گرد وائرہ لکھیں۔

int, pack, create, case, return, small, math, struct, program, library.

حل:

1. int
2. case
3. struct
4. return

مرکزی 1.4: C پرограм کے مختلف حصوں کی شناخت کریں۔

Identify different parts of the following C program:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    printf("I am a student of class 10th");
    getch();
}
```

حل:

#### **Header files**

```
#include<stdio.h>
#include<conio.h>
```

#### **Main section**

```
void main()
```

#### **Body of main function**

```
{
    printf("I am a student of class 10th" );
    getch();
}
```

حل:

مرکزی 1.5: درج زیل میں درست کمیٹس پر نشان لگائیں۔

Tick valid comments among the following:

- \*comment goes here\*
- /comment goes here/
- %comment goes here%
- /\* comment goes here \*/
- /\*comment goes here/
- //comment goes here \*/

/\*Comment goes here\*/

مرکزی 1.6: مندرجہ ذیل مستقلات کی اقسام کی شناخت کریں:

12	1.2	**	-21	32.768
'a'	-12.3	41	40.0	'\'

حل:

1. integer constant(12)
2. Real constant(1.2)
3. Character constant('\*)
4. Integer constant(-21)
5. Real Constant(32.768)
6. Character constant('a')
7. Real Constant (-12.3)
8. Integer constant(41)
9. Real Constant(40.0)
10. Character constant('\'')

مرکزی 1.7: درج ذیل میں سے تغیرات کے درست ناموں کے گرد دوسرہ لائیں۔

_Hello,	1var	roll_num	Air23Blue	float
Case	\$car	name	=color	Float

حل:

1. \_Hello
2. roll\_num
3. name
4. Air23Blue

## پروگرامنگ کا تعارف

19

سرگزی 1.8: ایک پروگرام لکھیں جو مناسب ڈینا پس کے متغیرات میں آپ کا ذاتی ڈینا محفوظ کرے۔ مندرجہ ذیل ڈینا کے ساتھ ان متغیرات کو شروع کریں:

- آپ کی عمر
- جنس کا پہلا حرف
- آپ کے آٹھویں جماعت کے نمبر

حل:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char n = 'M' , g = 'M';
    int age = 20,marks = 800;
    float height = 6.1;
    printf( " %c %c %d %d %f " , n , g , age , marks , height );
    getch();
    return 0;
}
```

## پروگرامنگ کا تعارف

### اضافی کشیر الاتجاحی سوالات

D	C	B	A	کشیر الاتجاحی سوالات	نمبر شمارہ
A اور B دونوں	ہارڈ دیسٹر	سافت دیسٹر	کمپیوٹر پروگرام	کمپیوٹر کو دی گئی بدایات کے سیٹ کو۔۔۔۔۔ کہا جاتا ہے	1
پروگرامنگ انوائرنمنٹ	کمپیوٹر پروگرامنگ	سافت دیسٹر	کمپیوٹر لینگوچ	کمپیوٹر میں ان بدایات کو لکھنے یا ذخیرہ کرنے کے عمل کو۔۔۔۔۔ کے نام سے جانا جاتا ہے۔	2
چور	تیکر	پروگرامر	ڈیلپہر	وہ شخص جو کمپیوٹر پروگرام کو صحیح طریقے سے لکھنا جانتا ہے اسے۔۔۔۔۔ کہتے ہیں۔	3
تمام	عربی	اردو	انگریزی	کمپیوٹر نہیں سمجھ سکتا۔	4
ان میں سے کوئی نہیں	کی - ورڈ	پروگرامنگ لینگوچ	کمپیوٹر پروگرام	پروگرام ان خصوصی لینگوچ میں کمپیوٹر پروگرام لکھتے ہیں جنہیں۔۔۔۔۔ کہا جاتا ہے۔	5
کمپیوٹر	ٹولز	سافت دیسٹر	ہارڈ دیسٹر	پروگرامر کو پروگرامنگ کے لیے۔۔۔۔۔ کی ضرورت ہوتی ہے۔	6
انٹیگریٹڈ ڈیلپہنٹ انوائرنمنٹ	انٹیگریٹڈ ڈیلپہنٹ انوائرنمنٹ	انٹر نیشنل ڈیلپہنٹ انوائرنمنٹ	انٹیگریٹڈ ڈیلپہنٹ انوائرنمنٹ	انٹیگریٹڈ ڈیلپہنٹ کی ضرورت ہے۔	7
انٹیگریٹڈ ڈیلپہنٹ انوائرنمنٹ	پروگرامنگ انوائرنمنٹ	ذخیرہ الفاظ	انٹر نیشنل ڈیلپہنٹ انوائرنمنٹ	ایک سافت دیسٹر جو پروگرامروں کو کمپیوٹر پروگرام لکھنے اور چلانے میں سہولت فراہم کرنے کے لیے ایک پروگرامنگ انوائرنمنٹ فراہم کرتا ہے اسے۔۔۔۔۔ کے نام سے جانا جاتا ہے	8
تمام	Code :: Blocks	X Code	Visual studio	کی پروگرامنگ لینگوچ کے لیے IDEs ہیں۔	9
سافت دیسٹر	کمپیوٹر	کمپائلر	بیکسٹ ایڈیٹر	۔۔۔۔۔ ایک سافت دیسٹر ہے جو پروگرامر کو کمپیوٹر پروگرام لکھنے اور ایڈٹ کرنے کی اجازت دیتا ہے۔	10
شاخت	بیکسٹ ایڈیٹر	IDE	کمپائلر	تمام IDEs کے اپنے مخصوص۔۔۔۔۔ ہوتے ہیں۔	11
صرف 0s	صرف 1s	1s اور 0s	حروف حجی	کمپیوٹر صرف ۔۔۔۔۔ پر مشتمل میں لینگوچ کو سمجھتا اور کام کرتا ہے۔	12

پر نظر	مشین	مترجم	کپاکٹر	لینگوچ میں لکھے ہوئے پروگرام کو مشین لینگوچ کوڈ میں تبدیل کرنے کا ذمہ دار ہے۔	13
سنیکس	ایر	انشر کشن	سافت ویئر	ہر پروگرامنگ میں کچھ ابتدائی تحریراتی عناصر ہوتے ہیں اور ایک درست پروگرام لکھنے کے لیے وہی اصول فراہم کرتا ہے۔ قواعد کا یہ مجموعہ لینگوچ کے --- نام کے طور پر جانا جاتا ہے۔	14
مشین ایر	پروگرامنگ ایر	منطقی ایر	سنیکس ایر	پروگرامنگ کے دوران اگر پروگرامنگ لینگوچ کے صحیح سنیکس یا قواعد پر عمل نہیں کیا جاتا ہے تو پروگرام مرتب نہیں ہوتا ہے۔ اس صورت میں کپاکٹر ایک غلطی پیدا کرتا ہے۔ اس قسم کی غلطیوں کو ----- کہتے ہیں۔	15
print	case	int	auto	مندرجہ ذیل میں سے کون سانظی کی ورثیتی ہے؟	16
5	4	3	2	سی پروگرام کی ساخت کے --- اہم حصے یہ ہیں۔	17
#include <header_ file_name>	#include (header_ file_name)	#include <head_ file_name>	#include <header_ file_name>	سنیکس کی عمومی ساخت اس طرح ہے: include	18
تمام	بڑی	ہیڈرفائلز	Main()	ہر سی پروگرام میں ایک ---- نقش ہوتا ہے اور یہاں سے پروگرام چلنے کا آغاز ہوتا ہے۔	19
()	{}	[]	↔	میں () کی بڑی ---- کے اندر ہوتی ہے۔	20
ہیڈر	ریزروورڈ	کینٹس	Syntax	ایک پروگرام میں وہ سنیکس ہیں جن کو کپاکٹر نظر انداز کرتا ہے اور ان پر عملدرآمد نہیں ہوتا ہے۔	21
ان میں سے کوئی نہیں	دونوں a اور b	ٹیکن لائن کینٹس	سنکل لائن کینٹس	کینٹس کی قسم ہیں۔	22

## پروگرامنگ کا تعارف

22

	//	/	:	سکل لائن کمینش شروع ہوتے ہیں۔	23
//	??	/.	~	ملی لائن کمینش شروع ہوتے ہیں۔	24
*/	/.	//	)	ملی لائن کمینش ---- پر ختم ہوتے ہیں۔	25
?	:	6	.	مندرجہ ذیل میں سے کوئی خاص علامت نہیں ہے؟	26
تمام	کی ورز	متغیرات	مستقلات	حروف صحیح ہندسے اور خاص علا میں جب قابل اجازت انداز میں مل جائیں تو۔	27
حقیقی مستقلات	مستقلات	اشیب مر مستقلات	کریکٹر مستقلات	---- وہ قسمیں ہیں جنہیں کسی پروگرام کے چلنے کے دوران تبدیل نہیں کیا جاسکتا۔	28
تمام	کریکٹر مستقلات	حقیقی مستقلات	اشیب مر مستقلات	مستقلات کی اقسام ہیں۔	29
متغیرات	کریکٹر مستقلات	حقیقی مستقلات	اشیب مر مستقلات	ایک اعشاریہ کے بغیر قیمت ---- ہے۔	30
-7941.2345?	-792345a	79412345	-7941.2345	مندرجہ ذیل میں سے کون حقیقی مستقلات ہے؟	31
کمینش	کی ورز	متغیر	مستقل	----- دراصل ایک میموری لوکیشن کو دیا گیا نام ہے کیونکہ ڈنیا کپیوٹر کی میموری کے اندر محفوظ ہوتا ہے۔	32
ان میں سے کوئی نہیں	کمینش	مستقلات	متغیر	پروگرام کی ایگزیکیوشن کے دوران ----- کی قیمت کو تبدیل کیا جاسکتا ہے۔	33
1	5	4	3	متغیر کی ڈنیا ناپ---- ہیں۔	34
ریکل ولیو	int	int ولیو	سائی ان int ولیو float	اشیب مر ڈنیا ناپ کو خیرہ کرنے کے لیے استعمال کیا جاتا ہے۔	35
8	6	4	2	اشیب مر میموری میں ---- بائس لیتا ہے	36

## پروگرامنگ کا تعارف

23

ان میں سے کوئی نہیں	a ورطہ دونوں	منفی قیمت	ثبت قیمت	ایک signed int شور کر سکتا ہے:	37
float	signed integer	unsigned integer	char integer	بلور ذیفالٹ ناپ int کو سمجھا جاتا ہے:	38
0 to -4,294,967,295	0 to $\pm 4,294,967,295$	0 to 2,112,222,222	0 to +4,294,967,295	کی حدود ہے: unsigned int	39
char data	int data	float data	real data	ٹائپ ایک حقیقی نمبر (فلوئنگ پاؤ اسٹ کے ساتھ نمبر) کو چند سوں تک درست رکھنے کے لیے استعمال کیا جاتا ہے۔	40
$3.4 \times 10^{-38}$ to $3.4 \times 10^{38}$	$4.4 \times 10^{-38}$ to $4.4 \times 10^{38}$	$3.4 \times 10^{-37}$ to $3.4 \times 10^{37}$	$4.4 \times 10^{-38}$ to $3.4 \times 10^{38}$	فلوئنگ ڈیٹا پاؤ اسٹ کی ولیوں میں ہے:	41
تمام	int salary;	float height;	unsigned int age;	درست متغیر ذیکریں کی مثال ہیں۔	42

جوابات:

1	2	3	4	5	6	7	8	9	10
D	C	B	D	B	C	A	D	D	A
11	12	13	14	15	16	17	18	19	20
C	B	A	D	A	D	B	A	A	C
21	22	23	24	25	26	27	28	29	30
B	C	C	B	D	B	D	C	D	A
31	32	33	34	35	36	37	38	39	40
A	B	A	A	C	B	C	C	A	B
41	D	42	D						

## پروگرامنگ کا تعارف

24

### اضافی مختصر سوالات

1: کپیوٹر پر د گرام کی وضاحت کریں؟

جواب: کپیوٹر کو دی گئی بدایات کے سینٹ کو کپیوٹر پر د گرام بساافت دیتے کہا جاتا ہے۔

2: کپیوٹر پر د گرامنگ کیا ہے؟

جواب: کپیوٹر میں بدایات کو لکھنے یا ذخیرہ کرنے کے عمل کو کپیوٹر پر د گرامنگ کہا جاتا ہے۔

3: پر د گرام کون ہے؟

جواب: جو شخص کپیوٹر پر د گرام کو صحیح طریقے سے لکھا جاتا ہے اسے پر د گرام کہا جاتا ہے۔

4: ی لینگوچ کس نے اور کب تیار کی؟

جواب: C لینگوچ دنس رچی نے 1969 اور 1973 کے درمیان میں لیبارٹریز میں تیار کی۔

5: پر د گرامنگ لینگوچ کیا ہے؟

جواب: پر د گرام ان خصوصی لینگوچز میں کپیوٹر پر د گرام لکھتے ہیں جنہیں پر د گرامنگ لینگوچ بوج کہا جاتا ہے۔ Python اور C++، C#، C، JAVA سب سے زیادہ استعمال ہونے والی پر د گرامنگ لینگوچ بوج ہیں۔

6: پر د گرامنگ انوائرنٹ کیا ہے؟

جواب: پر د گرامنگ کے لیے تمام ضروری ٹولز کا مجموعہ پر د گرامنگ کا انوائرنٹ بناتا ہے۔ پر د گرام لکھنے سے پہلے پر د گرامنگ انوائرنٹ قائم کرنا ضروری ہے۔ یہ ہمارے لیے پر د گرام لکھنے اور اس پر عمل کرنے کے لیے ایک بنیادی پیٹ فارم کے طور پر کام کرتا ہے۔

7: انٹیکٹر ڈی ڈی ڈیپٹنٹ انوائرنٹ (IDE) کیا ہے؟

جواب: سافت دیزائیز کپیوٹر پر د گرام لکھنے اور چلانے میں پر د گرامز کو سوالت دینے کے لیے پر د گرامنگ کا انوائرنٹ فراہم کرتا ہے اسے انٹیکٹر ڈی ڈی ڈیپٹنٹ انوائرنٹ (IDE) کہا جاتا ہے۔ IDE کا ایک گرافیکل یوزر انٹر فیس (Graphical User Interface) ہوتا ہے۔ اس کا مطلب یہ ہے کہ کوئی صارف ان پت فراہم کرنے اور آٹ پت حاصل کرنے کے لیے وندوز اور بنیوں کا استعمال کرتے ہوئے اس کے ساتھ باتیں چیت کر سکتا ہے۔

8: C پر د گرامنگ لینگوچ کے لیے IDEs لکھ دیں۔

جواب: پر د گرامنگ لینگوچ کے لیے کچھ IDEs یہ ہیں۔

Dev C++ (4)

Code :: Blocks (3)

X Code (2)

Visual studio (1)

9: نیکست ایڈیٹر کا مقدمہ کیا ہے؟ (ب) نیکست ایڈیٹر کیا ہے؟

جواب: نیکست ایڈیٹر ایک سافت دیزائیٹ ہے جو پر د گرامز کو کپیوٹر پر د گرام لکھنے اور ایڈٹ کرنے کی اجازت دیتا ہے۔ تمام IDEs کے اپنے خصوصی نیکست ایڈیٹر ہوتے ہیں۔

10: کمپائلر (Compiler) کی تعریف کریں؟

جواب: کمپائلر ایک سافت دیزائیٹ ہے جو ایک بول پر د گرامنگ لینگوچ میں لکھے ہوئے پر د گرام کو مشین لینگوچ کے کوڈ میں تبدیل کرنے کا ذمہ دار ہے۔

**11:** سینکس کی تعریف کریں؟

جواب: ایک درست پروگرام لکھنے کے لیے استعمال ہونے والے قواعد کا جمود لینگوچ کے سینکس کے طور پر جانا جاتا ہے۔ سینکس کو ایک پروگرام لینگوچ کا گرامر سمجھا جاسکتا ہے۔

**12:** سینکس ایرے کی امراد ہے؟

جواب: پروگرامنگ کے دوران اگر مناسب سینکس یا پروگرامنگ کے قواعد پر عمل نہیں کیا جاتا ہے تو پروگرام چلتا نہیں ہے۔ اس صورت میں کپاٹر ایک ایروڈیتا ہے۔ اس قسم کی غلطیوں کو سینکس ایر کہا جاتا ہے۔

**13:** آپ کی ورثے کے بارے میں کیا جانتے ہیں؟ (ب) کچھ ذخیرہ الفاظ کے نام لکھیں۔

جواب: ہر پروگرام لینگوچ میں الفاظ کی ایک فہرست ہوتی ہے جو پہلے سے بیان شدہ ہوتی ہے۔ ہر لفظ کے اپنے مخصوص معنی ہوتے ہیں جو پہلے سے ہی کپاٹر کو معلوم ہوتے ہیں۔ ان الفاظ کو مخصوص الفاظ یا کی۔ ورثے کے نام سے جانا جاتا ہے۔

مخصوص الفاظ کے کچھ نام درج ذیل ہیں: auto, int, else, switch and return

**14:** سینکوچ میں پروگرام کی ساخت کے اہم حصوں کے نام لکھیں؟

جواب: سی لینگوچ میں لکھے گئے پروگرام کو تین اہم حصوں میں تقسیم کیا جاسکتا ہے۔

main() .3 فناشن کی باڑی

2. من یکش

1. لک سیشن یا ہیڈر سیشن

**15:** لک سیشن یا ہیڈر سیشن کیا ہے؟

جواب: سی لینگوچ میں پروگرام لکھتے ہوئے ہم ان فناشز کا سچی استعمال کرتے ہیں جو پہلے ہی لینگوچ میں بیان کیے گئے ہیں۔ لیکن موجودہ فناشز کو استعمال کرنے سے پہلے ہمیں ان کو شامل کرنے کی ضرورت ہے جہاں ان فناشز کی وضاحت کی گئی ہو۔ ان فاکلوں کو ہیدرفائل کہا جاتا ہے۔ ہم ان ہیدرفائلوں کو پروگرام کے اوپری حصے میں include statement لکھ کر اپنے پروگرام میں شامل کرتے ہیں۔

**16:** سینکس کی عمومی ساخت لکھیں؟

جواب: #include<header\_file\_name> کی ساخت مندرجہ ذیل ہے:

**17:** سینکس کی وضاحت کریں؟

جواب: #include<header\_file\_name> کی ساخت مندرجہ ذیل ہے:

یہاں ہیدرفائل کا نام درج بالا مثال میں کسی بھی ہیدرفائل کا نام ہو سکتا ہے۔ ہم نے <facl.h> فائل کی ہے جس میں ان پٹ اور آؤٹ پٹ فناشز سے متعلق معلومات ہیں۔ بہت سی درسی ہیدرفائلیں بھی دستیاب ہیں مثلاً کے طور پر <math.h> میں تمام پہلے سے طے شدہ ریاضی کے فناشز شامل ہیں۔

**18:** سی لینگوچ میں من یکش کو بیان کریں؟

جواب: سی پروگرام میں ایک main() فناشن ہوتا ہے اور یہاں سے پروگرام چلانا شروع ہوتا ہے۔

**19:** من 0 یکش کی باڑی سے کیا ارادہ ہے؟

جواب: main() کی باڑی نیز ہی برکٹوں () (Curly Braces) کے اندر ہوتی ہے۔ تمام شیئنس جو نیز ہی برکٹوں () کے اندر موجود ہوتی ہیں "main" فناشن کی باڑی میں شامل ہوتی ہیں۔

20: ہی سینٹس میں پر و رام سنتے کے اصول ہیں؟

جواب: ہی لینگوچ پر و گرام میں سینٹس کی ترتیب اس ترتیب کے مطابق ہوئی چاہیے جس ترتیب سے ہم سینٹس کو چلانا چاہتے ہیں۔ ہی لینگوچ کیس سینٹس (Case Sensitive) ہے۔ اس کا مطلب یہ ہے کہ اگر کسی ذخیرہ لفظ کو تمام مجموعے حروف کے ساتھ بیان کیا جاتا ہے تو اسے بڑے حروف میں نہیں لکھا جاسکتا ہے لیکن int اور Int مختلف ہیں۔ جبکہ پہلا ذخیرہ الفاظ ہے اور دوسرا نہیں ہے۔ ہر سینٹ (Statement) کے آخر پر یہی کو ان (;) آتا ہے۔

21: کینٹس کیا ہیں؟

جواب: کینٹس ایک پر و گرام میں وہ سینٹس ہیں جن کو کپاٹل نظر انداز کرتا ہے اور ان پر عملدرآمد نہیں ہوتا ہے۔ عام طور پر کینٹس کو ذکی تفصیل فراہم کرنے کے لیے نظری لینگوچ میں لکھتے ہیں جیسے اگر یہی لینگوچ میں۔

22: کینٹس کیتھے کا مقصد کیسی؟

جواب: کینٹس کو پر و گرام کی دستاویزات کے طور پر سمجھا جاسکتا ہے۔ ان کا مقصد یہ ہے:

1) یہ دوسرے پر و گرامز کے کوڈ کو سمجھنے میں سہولت فراہم کرتا ہے۔

2) یہ ہمیں اپنے کوڈ کو لکھنے کے برسوں بعد بھی سمجھنے میں مدد کرتا ہے۔ ہم نہیں چاہتے کہ ان سینٹس پر عملدرآمد ہو کوئکد اس سے سینٹس کی غلطی ہو سکتی ہے کیونکہ سینٹس نظری لینگوچ میں لکھی ہوتی ہیں۔

23: کینٹس کی کتنی اقسام ہیں؟

جواب: ہی پر و گرام لینگوچ میں کینٹس کی دو اقسام ہیں۔ 1) سنکل لائے کینٹس 2) ملنی لائے کینٹس

24: سنکل لائے کینٹس ہوں ملنی لائے کینٹس میں کیا فرق ہے؟

جواب: سنکل لائے کینٹس // سے شروع ہوتے ہیں۔ // کے بعد کچھ بھی ایک ہی لائے پر لکھا ہو وہ کینٹ ہی سمجھا جاتا ہے۔ مثال کے طور پر  
// This is a comment.

ملنی لائے کینٹس // سے شروع ہوتے ہیں اور // پر ختم ہوتے ہیں۔ // اور // کے درمیان کسی بھی چیز کو ایک کینٹ سمجھا جاتا ہے چاہے ایک سے زیادہ سطروں پر ہوں۔ مثال کے طور پر:

/ • This is a Multi line

Comment • /

25: مستقلات کی تعریف کریں؟

جواب: مستقلات و قیسیں ہیں جنہیں پر و گرام کے پڑے کے دوران تبدیل نہیں کیا جاسکتا مثلا 1500, 75.7, 5, وغیرہ۔ ہی لینگوچ میں مستقلات کی تین اقسام ہیں: 1. شیبر مستقلات 2. حقیقی مستقلات 3. کریکٹر مستقلات

26: اشیب مسئلہ کی تعریف کریں؟

جواب: ان قیتوں میں اعشاریہ نہیں آتا ہے۔ یہ ثبت یا منفی ہو سکتی ہیں۔ مثلا 7, 1256, 55555, 30100, -54, 2349 وغیرہ۔ اگر قیمت سے پہلے کوئی علامت نہ ہوتے سے ثبت سمجھا جاتا ہے۔

27: حقیقی مسئلہ کی تعریف کریں؟

جواب:

حقیقی مسئلہ	کریکٹر مسئلہ
ان قیتوں میں اعشاریہ آتا ہے۔ مثلا 14.3333, 75.0, 15.3333 علمات یا پیشہ درڈ مستقل کریکٹر ہوتے ہیں جیسے '5', '7', 'X', 'a', '!' وغیرہ۔	11 کے اندر کوئی بھی چھوٹے یا بڑے حروف اچھوپ ایشن (Punctuation) کی علامت یا پیشہ درڈ مستقل کریکٹر ہوتے ہیں جیسے '5', '7', 'X', 'a', '!' وغیرہ۔

28: خنیر کیا ہے؟

جواب: خنیر دراصل میموری لوکیشن (Memory Location) کو دیا جانے والا نام ہے، کونکہ ڈنائیکسیورٹر کی میموری کے اندر محفوظ ہوتا ہے۔ ایک پروگرام میں خنیر کی قیمت کو تبدیل کیا جاسکتا ہے۔ اس کا مطلب یہ ہے کہ اگر کسی خنیر کی قیمت 5 پر مشتمل ہو تو بعد میں ہم اسے دوسری قیمت دے سکتے ہیں جو کہ قیمت 5 کو بدلتی ہے۔

مثال: height, average\_weight

29: خنیر کی کتنی ڈنائیکسیورٹر کیسیں ہیں؟

جواب: C لینگوچ میں استعمال ہونے والے ڈنائیکسی اقسام درج ذیل ہیں۔

1۔ اشیب	int	char	-3
---------	-----	------	----

2۔ فلوٹ پاؤٹ

3۔ float

30: اشیب کی وضاحت کریں؟

جواب: اشیب ڈنائیکسی اشیب و بیلووز کو محفوظ کرنے کے لیے استعمال کی جاتی ہے۔ ایک اشیب 4 بائیت میموری لیتا ہے۔ اشیب ڈنائیکسی خنیر کرنے کے لیے کی ورڈ لفظ int استعمال کرتے ہیں۔

31: unsigned int اور signed int کے درمیان کیا فرق ہے؟

signed int	unsigned int
Signed int میں -2,147,483,647 سے 2,147,483,647 تک تمام ثبت اور منفی دونوں قیمتیں محفوظ کی جاسکتی ہیں۔ بطور ڈنائیکسی int کو Signed int سمجھا جاتا ہے۔	ایک unsigned int صرف ثبت قیتوں کو محفوظ کر سکتا ہے اور اس کی قیمت 0 سے 4,294,967,295 تک ہوتی ہے۔ Unsigned int اشیب ڈنائیکسی کرنے کے لیے کی ورڈ int unsigned int استعمال کیا جاتا ہے۔

32: مخفیگ پا اکٹ فٹا کیا ہے؟

جواب: ڈینا اسپ ایک حقیقی نمبر (مخفیگ پا اکٹ کے ساتھ) افساریہ کے بعد زیادہ سے زیادہ پچ ہندسے آئتے ہیں۔ float اسپ کا خیر دیکھیر کرنے کے لیے ہم کی ورنہ float میوری کے 4 بائیس استعمال کرتا ہے۔ اس کی قیمت  $3.4 \times 10^{-38}$  سے  $3.4 \times 10^{38}$  تک ہوتی ہے۔

33: کریکٹر ڈینا اچھا کا تحدید کیا ہے؟

جواب: C میں کریکٹر ناچھ مخفیرات کو ذیکھیر کرنے کے لیے ہم کی ورنہ char استعمال کرتے ہیں۔ یہ ذخیرہ کرنے کے لیے صرف 1 بائیس میوری لیتا ہے۔

تمہارا خیر صرف ایک کریکٹر کو محفوظ کر سکتا ہے۔

34: مخفیرات کو ہم دینے کے لیے کوئی دو اصول لکھیں؟

جواب: ہر مخفیگ کا ایک منفرد نام یا شناخت کنندا ہونا ضروری ہے۔ مخفیگ کے نام کے لیے درج ذیل اصول استعمال کیے جاتے ہیں۔

(1) مخفیگ کے نام میں صرف حروف تہجی (بڑے یا چھوٹے) ہندے اور "۔" کی علامت آئکتی ہے۔

(2) مخفیگ کا نام ایک حرف یا اندر سکور ( ) سے شروع ہونا چاہیے یہ ہندے سے شروع نہیں ہو سکتا۔

35: مخفیرات اور مستقلات میں کیا فرق ہے؟

جواب: مخفیرات اور مستقلات کے درمیان فرق درج ذیل ہیں۔

مخفیرات	مستقلات
<p>مخفیگ در اصل میوری لوکیشن (Memory Location) کو دیا جانے والا نام ہے۔ کیونکہ ڈینا کپیوزری میوری کے اندر محفوظ ہوتا ہے۔ ایک پروگرام میں مخفیگ کی قیمت کو تبدیل کیا جاسکتا ہے۔ اس کا مطلب یہ ہے کہ کسی پروگرام میں اگر کسی مخفیگ کی قیمت 5 پر مشتمل ہو تو بعد میں ہم اسے دوسری قیمت دے سکتے ہیں جو کہ قیمت 5 کو بدلتی ہے۔</p> <p>مثال: height , average_weight</p>	<p>مستقلات وہ قیمتیں ہیں جنہیں پروگرام کے پلٹے کے دوران تبدیل نہیں کیا جاسکتا مثلا 1500, 5, 75.7 وغیرہ۔ سی لینگوچ میں مستقلات کی تین اقسام ہیں:</p> <p>1.1. شیبر مستقلات      2. حقیقی مستقلات</p> <p>3. کریکٹر مستقلات</p>

36: مخفیگ کو ذیکھیر بیش کیا ہے؟

جواب: پروگرام میں استعمال کرنے سے پہلے ہمیں مخفیگ کو ذیکھیر کرنے کی ضرورت ہے۔ مخفیگ کو ذیکھیر کرنے میں اس کی ڈینا اسپ کی وضاحت اور اسے ایک درست نام دینا شامل ہے۔ مخفیگ کو ذیکھیر کرنے کے لیے مندرجہ ذیل سنیکس کی ہدودی کی جاسکتی ہے۔

37: مخفیگ کی ایشلائر بیش کیا ہے؟

جواب: پہلی مرتبہ ایک مخفیگ کی قیمت سے منسوب کرنا مخفیگ ایشلائر بیش کہلاتا ہے۔ سی لینگوچ میں مخفیگ کو ذیکھیر کرتے ہوئے یا اس کے بعد ایشلائر کیا جاسکتا ہے۔

38: مخفیگ کی ایشلائر بیش کا طریقہ بیان کریں؟

جواب: مخفیگ کو ذیکھیر بیش کے وقت ایشلائر کرنے کے لیے عام سنیکس یہ ہے:

## یونٹ نمبر 2

## یوزر انٹریکشن

س: کپیور اور آئٹ پٹ فنکشن کیا ہے؟

کپیور ایک ایسا آلہ ہے جو ذینا کو ان پٹ کے طور پر لیتا ہے ذینا کو پرو سیس کرتا ہے اور آئٹ پٹ دیتا ہے۔ جبکہ آئٹ پٹ فنکشن پروگرام سے ذینا حاصل کرنے کے لیے استعمال ہوتا ہے۔ مانیٹر کے ذریعہ تیار شدہ آئٹ پٹ کو معیاری آئٹ پٹ کہا جاتا ہے۔ سی لینگوچ آئٹ پٹ غایبر کرنے کے لیے printf فنکشن میکری ہے۔

س: سی لینگوچ کے () printf فنکشن کو ایک مثال کے ساتھ بیان کریں۔ (یا) سکرین پر آئٹ پٹ غایبر کرنے کے لیے سی لینگوچ کا کون سا فنکشن استعمال کیا جاتا ہے؟

**printf()**

سکرین پر آئٹ پٹ دکھانے کے لیے سی پروگرامیں لینگوچ میں ایک بلٹ ان (built-in) فنکشن ہے۔ اس کا نام "پرنٹ فارمیٹ" printf (print formatted) سے لکھا ہے جو فارمیٹ شدہ آئٹ پٹ کو سکرین پر پرینٹ کرنے کے لیے استعمال ہوتا ہے۔ تمام ذینا کی اقسام printf فنکشن کے ساتھ غایبر کی جاسکتی ہیں۔ printf فنکشن کے کام کو سمجھنے کے لیے درج ذیل پروگرامیں مثال پر غور کریں:

مثال:

Program	Output:
#include<stdio.h> void main() { printf("Hello World"); }	Hello World

اس مثال میں printf فنکشن Hello World کو سکرین پر دکھانے کے لیے استعمال ہوتا ہے۔ ہم () فنکشن میں دو این (" ") کے اندر جو کچھ بھی لکھتے ہیں سکرین پر ظاہر ہوتا ہے۔

سوال: O / I (ان پٹ / آئٹ پٹ) اپر ہیزر میں فارمیٹ پیسٹار کی وضاحت کیوں اہم ہیں؟  
فارمیٹ پیسٹار ذینا اس فیلڈ کی چوڑائی اور سکرین پر ظاہر ہونے والے متغیر کی ولیوں کی محل کی نمائندگی کرتا ہے۔ ایک فارمیٹ پیسٹار ہیش علامت % سے شروع ہوتا ہے۔ فارمیٹ پیسٹار ان پٹ اور آئٹ پٹ دونوں شیئٹ کے لیے استعمال ہوتا ہے۔ فارمیٹ پیسٹار کا عمومی شیکس ہے۔ % مثال: اگر ہم کسی متغیر کی قیمت کو ظاہر کرنا چاہتے ہیں؟ ایک متغیر کوڈ یکسیر کریں اور پھر printf کے رویے کو چیک کریں۔

int age = 35;

printf ("age");

اب اگر ہم اس متغیر age کی قیمت سکرین پر ظاہر کرنا چاہتے ہیں۔ تو مندرجہ ذیل شیئٹ لکھیں گے:

لیکن یہ مقصد کو پرانیں کرتا کیونکہ یہ سکرین پر درج ذیل آئٹ پٹ دکھاتا ہے۔

یہ تغیر age کے اندر محفوظ شدہ قیمت کو ظاہر نہیں کرتا ہے اس کے بجائے یہ صرف وہی دکھاتا ہے جو printf کے وادیں کے اندر لکھا گیا تھا۔

درحقیقت ہمیں فارمیٹ سپیفایزر کی وضاحت کرنے کی ضرورت ہے جسے ہم سکرین پر دکھانا چاہتے ہیں۔ مندرجہ ذیل نیمیں یعنی گنج میں مختلف ڈیٹا ٹائپیں میں

متقابلے میں فارمیٹ سپیفایزر دکھاتا ہے۔

Data type	Format Specifier
int	%d or %i
float	%f
char	%c

فرض کریں کہ ہم int ڈیٹا ٹائپ کا لیٹ ایڈ کھانا چاہتے ہیں یعنی اسے printf کے اندر واضح کرنے کے لیے %d کا استعمال کرنا چاہیے۔ اسی

طرح float ڈیٹا ٹائپ کے لیے یعنی %f کا استعمال کرنا چاہیے۔ یہ مندرجہ ذیل مثال میں دکھایا گیا ہے:

### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    float height = 5.8;
    int age = 35;
    printf(" My age is %d and my height is %f ", age, height);
    getch();
}
```

### Output:

My age is 35 and my height is 5.800000

ہم دیکھ سکتے ہیں کہ آئٹ پٹ دکھاتے وقت پہلے فارمیٹ سپیفایزر کی جگہ وادیں کے بعد دیے گئے ڈیٹا میں سے پہلے تغیر یعنی age کی

قیمت آئی ہے اور دوسرا سے فارمیٹ سپیفایزر کی جگہ دوسرا تغیر آگیا ہے۔

جب ہم فلٹ ویجو ظاہر کرنے کے لیے %f کا استعمال کرتے ہیں تو یہ اعشاریہ کے بعد 6 ہندسوں کے بعد دکھاتا ہے۔ اگر ہم اعشاریہ کے بعد ہندسوں

کی تعداد بتانا چاہتے ہیں تو ہم nf % کو دیکھ سکتے ہیں جہاں n ہندسوں کی تعداد ہے۔ مندرجہ بالامثال میں اگر ہم یہ شیفت لکھیں۔

### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    float height = 5.8;
    int age = 35;
    printf(" My age is %d and my height is %.2f ", age, height);
    getch();
}
```

### Output:

My age is 35 and my height is 5.80

## یوزر انٹریکشن

31

فارمیٹ پیغماڑ صرف تغیر کے لیے استعمال نہیں ہوتے۔ دراصل ان کا استعمال تغیرات، کائنٹنٹ یادوں پر مشتمل کسی بھی ایکسپریشن کا نتیجہ ظاہر کرنے کے لیے کیا جاتا ہے جیسا کہ نیچے دی گئی مثال میں دکھایا گیا ہے۔

### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("Sum of 23 and 45 is %d", 23 + 45);
    getch();
}
```

### Output

Sum of 23 and 45 is 68

س: **scanf()** پر پٹ کیسیں؟

**scanf()** سی لینکوگ میں ایک بلک ان فنکشن ہے جو صارف سے ویریزبلز میں ان پٹ لیتا ہے۔ **scanf** فنکشن میں ہم فارمیٹ پیغماڑ کی مدد سے ان پٹ ذیلا کی متوقع قسم کی وضاحت کرتے ہیں۔ اگر صارف اٹیجگر ذیلا پٹ میں ذیادت ہے تو **scanf** میں بیان کردہ فارمیٹ پیغماڑ **d** یا **%** ہونا چاہیے۔ مندرجہ ذیل مثال پر غور کریں۔

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    char grade;
    scanf("%c", &grade);
    getch();
}
```

اس مثال میں **c** فارمیٹ پیغماڑ کر کیکٹر ناٹپ ویری ابیل کے لیے استعمال کیا گیا ہے۔ صارف کی دی گئی ان پٹ کو تغیر **grade** میں محفوظ کیا گیا ہے۔ **scanf** فنکشن کے دو اہم حصے ہیں جس کو اپر کوڈ میں دیکھا جاسکتا ہے۔ دوں کے اندر پہلا حصہ فارمیٹ پیغماڑ کی فہرست ہے اور دوسرا حصہ تغیرات کی فہرست ہے اور ان کے باس طرف & لکھا جاتا ہے۔

### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int number;
    printf("Enter a number between 0-10:");
    scanf("%d", &number);
    printf("The number you entered is: %d", number);
    getch();
}
```

### Output

Enter a number between 0-10: 4  
The number you entered is : 4

ہم ایک ہی `scanf` فنکشن کا استعمال کرتے ہوئے متعدد ان پت لے سکتے ہیں۔ مندرجہ ذیل بیان پر غور کریں۔

`scanf ("%d%d%d", &a, &b, &c);`

یہ دو شیعہ ویری اینل a اور b اور ایک فلٹ ویری اینل c میں ان پت لیتا ہے۔ ہر ان پت کے بعد صارف کو ایک space یا enter دباتا ہے۔ تمام ان پت کے بعد صارف کو enter دبالتا ہے۔ `scanf` فنکشن میں & اور پریٹ کو بھول جانا بہت عام ہی غلطی ہے۔ & اور پریٹ کے بغیر پروگرام ایگزکیوٹ تو جاتا ہے مگر متوقع مانع نہیں دیتا ہے۔

سوال: () پر نوٹ لکھیں؟  
`getch()`

`getch()` فنکشن صارف سے ایک کریکٹر لینے کے لیے استعمال کیا جاتا ہے۔ صارف کا داخل کردہ کریکٹر سکرین پر ظاہر نہیں ہوتا ہے۔ یہ فنکشن عام طور پر پروگرام کی ایگزکیوٹ کو روکنے کے لیے استعمال کیا جاتا ہے کیونکہ پروگرام اس وقت تک آگے نہیں چلا جب تک کہ صارف کوئی نہ دیتا۔ اس فنکشن کو استعمال کرنے کے لیے ہمیں لا بیربری `conio.h` کو پروگرام کے ہدایت سکشنس میں شامل کرنے کی ضرورت پڑتی ہے۔

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("Enter any key if you want to exit program");
    getch();
}
```

درج بالا پروگرام صارف سے کوئی بھی کریکٹر ان پت لیتا ہے اور پھر پروگرام کے عمل کو مکمل کرنے سے پہلے صارف کے ان پت کا انتظار کرتا ہے۔

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    char key;
    printf("Enter any key:");
    key = getch(); // Gets a character from user into variable key
    getch();
}
```

اگر ہم اس پروگرام کو چلاتے ہیں تو ہمیں `scanf` کا استعمال کرتے ہوئے ایک کریکٹر کو پڑھنے اور `getch` فنکشن کا استعمال کرتے ہوئے ایک کریکٹر کو پڑھنے میں فرق نظر آتا ہے۔ جب ہم `scanf` کے ذریعے کریکٹر پڑھنے ہیں تو ہمیں پروگرام کو آگے چلانے کے لیے enter دبانے کی ضرورت ہوتی ہے۔ لیکن `getch` کی صورت میں یہ enter کی کو دبانے کا انتقاد نہیں کرتا ہے۔ فنکشن ایک کریکٹر کو پڑھتا ہے اور اگلی لائن کو چلانے کی طرف بڑھتا ہے۔

س: () پر نوٹ لکھیں؟  
clrscr()

یہ فناشن سی لینگوچ ایڈیٹر کی آؤٹ پٹ سکرین کو صاف کرنے کے لیے استعمال ہوتا ہے۔

مثال:

### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int number;
    printf("Enter a number between 0-10:");
    scanf("%d", &number);
    printf("The number you entered is: %d", number);
    getch();
}
```

### Output

Enter a number between 0-10: 4
The number you entered is : 4

س: سینٹرٹر مینیٹر کیا ہے؟

سینٹرٹر مینیٹر (Statement Terminator)

سینٹرٹر مینیٹر کا پالٹر کے لیے شاخت کنندہ ہے جو لائن کے اختام کی نشاندہ کرتا ہے۔ سی لینگوچ میں سی کو ان (;) بطور سینٹرٹر مینیٹر استعمال ہوتا ہے۔ اگر ہر سینٹرٹر کے آخر میں سی کو ان نہ لگائیں تو ایرر (Error) آ جاتا ہے۔

printf("Hello World") (;) → Statement  
Terminator

س: اسکیپ سیکو ننس کیا ہیں؟ ہمیں ان کی ضرورت کیوں ہے؟

اسکیپ سیکو ننس (Escape Sequence)

اسکیپ سیکو ننس printf فناشن میں داوین ("") کے اندر استعمال ہوتے ہیں۔ جو printf کو استعمال کرتے ہوئے آؤٹ پٹ دکھانے کے اپنے

printf ("My name is \"Ali\"");

طریقہ کار کو تبدیل کرنے پر مجبور کرتے ہیں۔ مثال:

**My name is "Ali"**

مندرج بالا سینٹرٹر کی آؤٹ پٹ یہ ہے۔

مندرج بالا مثال میں " " ایک اسکیپ سیکو ننس ہے۔ یہ printf کو "کپیوڑا سکرین پر" ظاہر کرنے کا عب بنتا ہے۔

اسکیپ سیکو ننس کی بنا پر

اسکیپ سیکو ننس دو حروف پر مشتمل ہے۔ پہلا کریکٹر بیسٹ بیک سلیش (back slash) ("") ہوتا ہے اور دوسرا کریکٹر مطلوب فونکشنیٹی کے مطابق آتا ہے جسے ہم حاصل کرنا چاہتے ہیں۔ بیک سلیش (\) کو اسکیپ کر کریکٹر کہا جاتا ہے جو اسکیپ کے بعد میں نشاندہی کرنے کے لیے ہر اسکیپ سیکو ننس سے مغلک ہوتا ہے۔ اسکیپ کر کریکٹر اور اس کے بعد آنے والے کریکٹر سکرین پر ظاہر نہیں ہوتے ہیں لیکن یہ مخصوص کام کرتے ہیں جو ان سے منسوب کیا گیا ہے۔ درج ذیل اسکیپ سیکو ننس عام طور پر C لینگوچ میں استعمال ہوتے ہیں۔

مقدار	سیکو ننس	مقدار	سیکو ننس
الرٹ کی آواز پیدا کرتا ہے	\a	سنکل کوت سکرین پر دکھانا (\)	\
پچھلا کریکٹر مٹاتا ہے	\b	بیک سلیش سکرین پر دکھانا (\)	\।
اگلی لائن پر لے کر جاتا ہے	\n	8 پیس سکرین پر دکھانا	\t

نولائن (\n) (New line)

اسکیپ کر کریکٹر کے بعد n واضح کرتا ہے کہ کرسر (cursor) کو اگلی لائن (New Line) کے شروع پر لے کر جانا ہے۔ یہ اسکیپ سیکو ننس ایک سے زیادہ لائنوں پر آؤٹ پٹ پر نہ کرنے کے لیے استعمال ہوتا ہے۔ مثال:

#### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("My name is Ali. \n");
    printf("I live in Lahore");
    getch();
}
```

#### Output

My name is Ali.  
I live in Lahore

اسکیپ سیکو ننس کے بغیر اگر ہمارے پاس ایک سے زیادہ printf سینٹنٹ بھی ہوں تو ان کی آؤٹ پٹ ایک ہی لائن پر ظاہر ہو گی۔ مندرجہ ذیل مثال یہ بات واضح کرتی ہے۔ مثال:

ذیل مثال یہ بات واضح کرتی ہے۔ مثال:

#### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("My name is");
    printf("Ahmad");
    getch();
}
```

#### Output

My name is Ahmad

\t "شیب"

اس کیپ سیکونس \t , \n ایک فنکشن کو بتاتا ہے کہ افتنی طور پر اگلے نیب سنپ (Tab Stop) پر جانا ہے۔ ایک شیب اسٹاپ 8 کیسے کا مجموعہ ہوتا ہے۔ \t کا استعمال کر سر کو اگلے شیب اسٹاپ پر لے جاتا ہے۔ مثال:

**Program**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("Name: \tAli\nFname: \tHamad\nMarks: \t1000");
    getch();
}
```

**Output**

Name: Ali
Fname: Hamad
Marks: 1000

س: آپٹر ز کیا ہیں؟

(Operators)

آپٹر ز ریاضی کی علامتیں ہیں جو اپرینڈ (operator) پر کچھ آپریشن کرتی ہیں۔ اپرینڈ ز دیری ای بلز یاد طیوڑ ہو سکتی ہیں۔ C میں بہت سے آپٹر ز استعمال ہوتے ہیں ان میں سے کچھ یہ ہیں:

- اسائنسٹ آپٹر (Assignment operator)
- ار تمیلک آپٹر (Arithmetic operators)
- متعلق آپٹر (Logical operators)
- رلیشنل آپٹر (Relational operators)

س: اسائنسٹ آپٹر کی وضاحت کریں؟

اسائنسٹ آپٹر (Assignment Operator)

اسائنسٹ آپٹر ایک تغیر کی قیمت تفویض کرنے کے لیے استعمال کیا جاتا ہے یا ایک تغیر کی قیمت کو دوسرے تغیر میں رکھنے کے لیے استعمال ہوتا ہے۔ برابری کی علامت (=) کوئی لینگوچ میں اسائنسٹ آپٹر کے طور پر استعمال کیا جاتا ہے۔ مثال:

کوڈ کی یہ لائن پڑھنے کے بعد دیری ای بل sum میں قیمت 5 محفوظ ہو جائے گی۔

int sum = 6;

int var = sum;

پہلے دیری ای بل sum کی قیمت 6 رکھی گی بہر اگلی لائن میں sum کی قیمت دیری ای بل var سے منسوب کر دی گئی۔

مثال: ایک پروگرام کھص جو دو اشیج ویری لے بلز کی قیتوں کا آپس میں تبدل (swap) کرے۔

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int a = 2, b = 3, temp;
    temp = a;
    a = b;
    b = temp;
    printf("Value of a after swapping: %d\n", a);
    printf("Value of b after swapping: %d\n", b);
    getch();
}
```

س: کون سے آپریٹر ارتمیٹک آپریٹر کے لیے استعمال ہوتے ہیں؟

ارتمیٹک آپریٹر (Arithmetic Operators)

ارتمیٹک آپریٹر زبان پر ریاضی کے کام کرنے کے لیے استعمال ہوتے ہیں۔ مندرجہ ذیل جدول ارتمیٹک آپریٹر کو ان کی تفصیل کے ساتھ پیش کرتا ہے۔

آپریٹر	نم	تفصیل
/	تھیم کا آپریٹر	یہ بائیں طرف کی قیمت کو دائیں طرف کی قیمت سے تقسیم کرنے کے لیے استعمال ہوتا ہے۔
*	ضرب کا آپریٹر	یہ دو قیتوں کو ضرب دینے کے لیے استعمال ہوتا ہے۔
+	جمع کا آپریٹر	یہ دو قیتوں کو جمع کرنے کے لیے استعمال ہوتا ہے۔
-	تفریق کا آپریٹر	یہ بائیں طرف کی قیمت کو دائیں طرف سے تفریق کرتا ہے۔
x	ماڈولس کا آپریٹر	یہ بائیں طرف کی قیمت کو دائیں طرف کی قیمت پر عددی تقسیم کرنے کے بعد بچنے والی قیمت دیتا ہے۔

## تقسیم (Division)

تقسیم کا آپریٹر (/) بائیں اور بینڈ کی قیمت کو دائیں اور بینڈ کی قیمت پر تقسیم کرتا ہے۔ میں: float result = 3.0 / 2.0; سینت چلنے کے بعد متغیر کا نتیجہ 1.5 آتا ہے۔ اگر دونوں اور بینڈ انسیجر ناپ ہیں تو تقسیم کا نتیجہ بھی int ناپ کا ہو گا۔ انسیجر کا جواب دینے کے لیے باقی نظر انداز کر دیا جاتا ہے۔ کوڈ کی مندرجہ ذیل لائن پر غور کریں۔

چونکہ دونوں قیمتیں int ناپ کی ہیں لہذا جواب بھی ایک انسیجر ہو گا جو 1 ہے۔ جب یہ قیمت (1) کو فلوٹ ناپ کے متغیر result سے منسوب کیا جائے گا تو یہ (1) فلوٹ ناپ میں تبدیل ہو جاتا ہے۔ لہذا اولیو 1.0 کو متغیر result میں حفظ کیا جاتا ہے۔ اگر ہم درست اور جامع جواب حاصل کرنا چاہتے ہیں تو پھر اور بینڈ ایک فلوٹ ناپ کا ہونا چاہیے۔ کوڈ کی مندرجہ ذیل لائن پر غور کریں:

float result = 3.0 / 2;

مندرجہ بالامثال میں متغیر result کی قیمت 1.5 ہے۔

مثال: ایک پروگرام لکھیں جو جاکلیٹ کے ایک ڈب اور جاکلیٹ کی تعداد جو اس ڈب میں ہے کی ان پڑتے۔ اور پروگرام ایک چاکلیٹ کی قیمت دکھائے۔

**Program**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    float box_price, num_of_chocolates, unit_price;
    printf("Please enter the price of whole box of chocolates:");
    scanf("%f ", &box_price);
    printf("Please enter the number of chocolates in the box:");
    scanf("%f ", &num_of_chocolates);
    unit_price = box_price / num_of_chocolates;
    printf("The price of a single chocolates is: %f ", unit_price);
    getch();
}
```

**Output**

```
Please enter the price of whole box of chocolates: 150
Please enter the number of chocolates in the box: 50
The price of a single chocolates is: 3.000000
```

## ضرب (Multiplication)

```
int multiply = 5 * 5;
```

ضرب آپریٹر ( $*$ ) ایک باتنزی آپریٹر ہے جو دو نمبروں کو ضرب دتتا ہے۔ جیسے:

اس سینٹنٹ پر عمل درآمد کے بعد ویری اینٹل (multiply) کی قیمت 25 ہو جائے گی۔

مثال: ایک پروگرام لکھیں جو مستطیل کی لمبائی اور چوڑائی بطور ان پٹ لے اور سکرین پر مستطیل کا رقم دکھائے۔

### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    float length, width, area;
    printf("Please enter the length of rectangle:");
    scanf("%f", &length);
    printf("Please enter the width of rectangle:");
    scanf("%f", &width);
    area = length * width;
    printf("Area of rectangle is : %f", area);
    getch();
}
```

### Output

```
Please enter the length of rectangle: 6.5
Please enter the width of rectangle: 3
Area of rectangle is: 19.500000
```

## جمع (Addition)

```
int add = 10 + 10;
```

جمع کا آپریٹر ( $+$ ) دو اوپرینڈ کو جمع کرتا ہے۔ جیسے:

عنصر add کی قیمت 20 ہو جائے گی۔

مثال: ایک پروگرام لکھیں جو دو مضمومین کے نمبرز یو زر سے ان پٹ لے اور ان نمبروں کو جمع کر کے سکرین پر دکھائے۔

### Program

```
#include <stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int sum, math, science;
    printf("Enter marks of Mathematics:");
    scanf("%d", &math);
    printf("Enter marks of Science:");
    scanf("%d", &science);
    sum = math + science;
    printf("Sum of marks is : %d", sum);
    getch();
}
```

### Output

```
Enter marks of Mathematics: 90
Enter marks of Science: 80
Sum of marks is: 170
```

شیئنٹ;  $a = a + 1$  ویری ابج a کی قیمت میں 1 کا اضافہ کرنے کے لیے استعمال ہوتی ہے۔ C لینگوچ میں اس شیئنٹ کو; ++a یا a++ بھی لکھا جاسکتا ہے۔ اسی طرح; --a یا a-- ویری ابج a کی قیمت میں 1 کو کم کرنے کے لیے استعمال کیا جاتا ہے۔

## تفریق (Subtraction)

**int result = 20 - 15;** تفریق کا آپریٹر (-) بائیں اور پرینڈ سے دائیں اور پرینڈ کو تفریق کرتا ہے۔ جیسے: تفریق کرنے کے بعد متغیر result کی قیمت 5 ہو جاتی ہے۔

## ماڈولس آپریٹر (Modulus Operator)

ماڈولس آپریٹر (%) بائیں اور پرینڈ کو دائیں اور پرینڈ پر تقسیم کرتا ہے اور تقسیم کے بعد بچنے والی باقی رقم لوٹاتا ہے۔ ماڈولس آپریٹر انجینئرنگ میں بھی کام کرتا ہے۔ جیسے:

**int remaining = 14 % 3;** اگر ہم 14 کو 3 سے تقسیم کریں تو 2 باقی رکھے گا۔ اس لیے ویری ابج remaining کی قیمت 2 ہو گی۔ مثال: ایک پروگرام لکھیں جو کسی ان پڑ کیے گے نمبر کا ماڈولس آپریٹر استعمال کرتے ہوئے صحیح عدد معلوم کرے۔

### Program

```
#include <stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int num, digit;
    printf("Enter a number:");
    scanf("%d", &num);
    digit = num % 10;
    printf("Right most digit of number you entered is: %d", digit);
    getch();
}
```

### Output

Enter a number: 789
Right most digit of number you entered is: 9

اگرچہ C لینگوچ میں ار تھیٹک شیئنٹس لکھتے ہوئے ایک عام غلطی الجبرا کے روزمرہ کے قوانین کو استعمال کرتا ہے۔ مثلاً  $y^6$  کو  $y^6$  لکھنا اور  $x^0 \times x^3$  کو  $x^3$  وغیرہ۔ اس کے نتیجے میں کپاٹلائرر (Compiler Error) آتا ہے۔

س: بر میشل آپریٹر کیا ہیں؟ ایک مثال کے ساتھ بیان کریں۔

## ریلیشن آپریٹر (Relation Operator)

ریلیشن آپریٹر زد و قیتوں کے درمیان تعلق کا تعین کرنے کے لیے انکا موازنہ کرتے ہیں۔ ریلیشن آپریٹر شناخت کرتے ہیں کہ قیمتیں برابر ہیں یا برابر نہیں۔ ایک قیمت دوسری قیمت سے بڑی ہے یا چھوٹی ہے۔ سی لینگوچ میں یونیورسک اور char ناچ کے ذمہ پر ریلیشن آپریٹر زمانہ دینے کی اجازت دیتی ہے۔ بنیادی ریلیشن آپریٹر ان کی تفصیل کے ساتھ ذیل میں دیے گئے ہیں:

تفصیل	ریٹائل آپریٹر
کے برابر ہے	$= =$
کے برابر نہیں ہے	$! =$
سے زیادہ	$>$
سے کم	$<$
سے زیادہ برابر	$> =$
سے کم برابر	$< =$

ریٹائل آپریٹر دو اور یہ پر آپریٹشن انجام دیتے ہیں اور یہ میں ایکپریشن (صحیح یا غلط) میں نتیجہ دیتے ہیں۔ اگر ایک قیمت صحیح (True) ہو تو قیمت 1 ہو گی اور اگر غلط (False) ہو تو قیمت 0 سے ظاہر کی جاتی ہے۔

جواب	تفصیل	ریٹائل ایکپریشن
True	? 1 برابر ہے 5	$5 == 5$
True	? 1 برابر نہیں ہے 7	$5 != 7$
False	? 7 سے 5 برابر ہے	$5 > 7$
True	? 5 چھوٹا ہے 7 سے	$5 < 7$
True	? 5 برابر ہے 5 یا برابر ہے 5	$5 > = 5$
False	? 5 چھوٹا ہے یا برابر ہے 4 کے لئے	$5 < = 4$

س: اسائنس آپریٹر ( $=$ ) اور برابر کے آپریٹر ( $= =$ ) میں کافر فرق ہے؟

اسائنس آپریٹر ( $=$ ) اور برابر کے آپریٹر ( $= =$ ) میں فرق

سی لینگوچ میں ( $= =$ ) آپریٹر دو ایکپریشن کی مساوات کی جانچ کے لیے استعمال ہوتا ہے کہ وہ برابر ہیں کہ نہیں۔ جبکہ ( $=$ ) آپریٹر دائیں والی ایکپریشن کو بابیں والے ویری ایبل سے منسوب کرتا ہے۔ ذہن ایکوئل (equal) آپریٹر ( $= =$ ) چیک کرتا ہے کہ دونوں طرف یہ آپریٹر زبردیں یا نہیں۔

سنگل ایکوئل آپریٹر ( $=$ ) دائیں اور یہ پر کو بابیں اور یہ پر کو بابیں سے منسوب کرتا ہے۔

ہم printf فونکشن کو ریٹائل ایکپریشن کا نتیجہ دکھانے کے لیے بھی استعمال کر سکتے ہیں۔ مثلاً

```
printf("%d", 5 == 5); // This statement displays 1
```

```
printf("%d", 5 > 7); // This statement displays 0
```

س: منطقی آپریٹرز کیا ہیں؟ مثال کے ساتھ بیان کریں۔

### منطقی آپریٹرز (Logical Operators)

منطقی آپریٹرز بولین ایکپریشن پر آپریشن سر انجام دیتے ہیں اور اس کے نتیجے میں جواب بھی بولین ایکپریشن ہوتا ہے۔ ریلیشنل آپریشن کا جواب بولین ایکپریشن ہے۔ ایک سے زیادہ ریلیشنل ایکپریشن کا جائزہ لینے کے لیے منطقی آپریٹرز کو سر انجام دیا جاسکتا ہے۔ مندرجہ ذیل نیادی منطقی آپریٹرز اور ان کی تفصیل کو ظاہر کرتا ہے۔

تفصیل	آپریٹر
Logical AND	&&
Logical OR	
Logical NOT	!

### آپریٹر AND (&&)

آپریٹر AND (&&) دو بولین ایکپریشن کو بطور اور پرینڈ لیتا ہے اور اگر اس کے دونوں ایکپریشن درست (true) ہوں تو نتیجہ درست آتا ہے۔ اگر کوئی بھی ایکپریشن غلط (false) ہو تو جواب غلط آتا ہے۔ AND آپریٹر کے لیے خود تھے نیبل ذیل میں دکھایا گیا ہے۔

رزلٹ	اکپریشن 2	آپریٹر	اکپریشن 1
False	False	&&	False
False	True	&&	False
False	False	&&	True
True	True	&&	True

### آپریٹر OR (||)

آپریٹر OR دو بولین ایکپریشن لیتا ہے اور کم از کم ایک ایکپریشن درست (true) ہونے پر جواب درست دیتا ہے۔ OR آپریٹر کے لیے خود تھے نیبل ذیل میں دکھایا گیا ہے:

رزلٹ	اکپریشن 2	آپریٹر	اکپریشن 1
False	False		False
True	True		False
True	False		True
True	True		True

آپریٹر (!) NOT

آپریٹر بولتے ایک پریشن کی قیمت کی نفی یا الٹ کرتا ہے۔ یہ اسے درست (true) بنا دتا ہے اگر یہ غلط ہے اور اگر یہ غلط (false) ہے تو اسے درست بنا دتا ہے۔ آپریٹر کے لیے نو تھہ نیجل ذیل میں دیا گیا ہے۔

رد	آپریٹر	ایک پریشن
True	!	False
False	!	True

منطقی آپریٹر کی مثالیں:  
مندرجہ ذیل نیجل مثالوں کی مدد سے منطقی آپریٹرز کے تصور کو ظاہر کرتا ہے۔

جواب	تفصیل	منطقی آپریٹر
False	چھوٹا ہے 4 سے 8 تک AND ? 3 < 4 && 7 > 8	
True	1 ہے 3 OR ? 3 == 4    3 > 1	
False	2 ہے 4 OR ? !(4 > 2    2 == 2)	
True	6 چھوٹا ہے 2 سے 6 ? 6 < = 6 && !(1 > 2)	
True	9 ہے 8 OR ? 8 > 9    !(1 < = 0)	

سی لینکوون شارکٹ ایلو جوئیشن (Short circuit evaluation) کرتی ہے۔ اس کا مطلب ہے کہ:

- اگر AND آپریٹر کو حل کرتے ہوئے ایک پریشن آپریٹر کی بائیں طرف والا حصہ غلط (False) ہو تو پورا ایک پریشن حل کیے بغیر جواب غلط ہو جاتا ہے۔
- اگر OR آپریٹر کو حل کرتے ہوئے ایک پریشن آپریٹر کی بائیں طرف والا حصہ درست (true) ہو تو پورا ایک پریشن حل کیے بغیر جواب درست ہو جاتا ہے۔

یو زری آپریٹر اور ہائزری آپریٹر میں کیا فرق ہے؟

یو زری بمقابلہ ہائزری آپریٹر (Unary vs Binary Operator)

تمام آپریٹرز کو دو بنیادی اقسام میں تقسیم کیا جاسکتا ہے جس کی بنیاد پر آپریٹر کو لاگو کیا جاسکتا ہے۔

یو زری آپریٹر (Unary Operator)

یو زری آپریٹر صرف ایک آپریشن پر لاگو ہوتے ہیں جیسے منطقی NOT (!) آپریٹر کے پاس صرف ایک آپریشن ہوتا ہے۔ سائنس آپریٹر (-)

یو زری آپریٹر کی ایک اور مثال ہے۔ جیسے 5 -

### بائنزی آپریٹر (Binary Operator)

بائنزی آپریٹر کو آپریشن کرنے کے لیے دو ایک پریشن کی ضرورت ہوتی ہے۔ مثلاً تمام ار تھیٹنک اور سینٹشل آپریٹر بائنزی آپریٹر ہیں۔  
مطلق آپریٹر && اور || بائنزی آپریٹر ہیں۔

### ترنی آپریٹر (Ternary Operator)

سی پروگرامنگ لینگوچ میں ایک ٹرنی آپریٹر (Ternary Operator) ہے جو تم اپریٹر پر کام کرتا ہے۔  
س: آپریٹر کی ترجیح سے کامرا دے؟ کون سی آپریٹر کو C زبان میں سب سے زیادہ فوکس حاصل ہے؟

### آپریٹر کی ترجیح (Operator Precedence)

اگر ایک پریشن میں ایک سے زیادہ آپریٹر ہیں تو سوال یہ پیدا ہوتا ہے کہ پہلے کسے حل کریں گے۔ اس مسئلے کو حل کرنے کے لیے ہر آپریٹر کو ترجیح دی گئی ہے۔ جس آپریٹر کی ترجیح زیادہ ہے وہ دوسرے آپریٹر سے پہلے حل ہو گا۔ اگر آپریٹر کی ایک جیسی ترجیح ہو تو دو ایک طرف والے آپریٹر سے پہلے باسیں طرف والا آپریٹر حل کریں گے۔

ترجیح	آپریٹر
1	()
2	!
3	*, /, %
4	+, -
5	>, <, >=, <=
6	==, !=
7	&&
8	
9	=

مثال:  $(5 * 4) / 2 * 3 + 7 \% 3 + 20$  کو حل کریں۔

$= 18 / 2 * 3 + 7 \% 3 + (5 * 4);$	()	پہلے بریکٹ حل ہو گی
$= 18 / 2 * 3 + 7 \% 3 + 20;$	/	دوسرے نمبر پر تقسیم کا عمل ہو گا
$= 9 * 3 + 7 \% 3 + 20;$	*	پھر ضرب کا عمل ہو گا
$= 27 + 7 \% 3 + 20;$	%	پھر ماڈولس آپریٹر حل ہو گا
$= 27 + 1 + 20;$	+	پھر جمع کا عمل ہو گا
$= 28 + 20;$	+	پھر جمع کا عمل ہو گا
$= 48;$		

## شکن

سوال نمبر 1: کشیر الائچی سوالات۔

نمبر	سوالات	D	C	B	A
1	printf( ) کا ڈیپرمنٹ کرنے کے لیے استعمال ہوتا ہے۔	تمام	char	float	int
2	C scanf _____ پروگرام کیونکی میں ہے۔	کوئی بھی نہیں	فائل	لاجبری	مطلوبہ لفظ
3	ان پت لینے کے لیے استعمال ہوتا ہے۔ getch()	تمام	char	float	int
4	کوڈ (code) کا یہ حصہ ایگزیکوٹ ہونے بعد صاف a کی قیمت کیا ہو گی؟ int a = 4; float b = 2.2; a = a * b;	8.2	8.0	8	8.8
5	مندرجہ ذیل میں سے کون سی کوڈ کی درست لائن ہے؟	کوئی بھی نہیں	line = this is a line;	grade = 'A';	int = 20;
6	ان میں سے کس آپریٹر کی ترجیح سب سے زیادہ ہے؟	!	>	=	/
7	مندرجہ ذیل میں سے کون سی آپریٹر کی قسم نہیں ہے:	لاجبل آپریٹر	ریٹیٹل آپریٹر	چیک آپریٹر	ارٹیکل آپریٹر
8	آپریٹر _____ کا حساب لگانے کے لیے استعمال ہوتا ہے۔	مراع	فائل	رمینڈر	پر شیج
9	مندرجہ ذیل میں سے کون سا کریکٹری لینگوچ میں درست ہے:	کوئی بھی نہیں	'9'	"a"	"here"
10	سی لینگوچ کے بارے میں کوئی آپشن درست ہے؟	کوئی بھی نہیں	تمام لا جبل آپریٹر آپریٹر زبانیں۔	نام کے طور پر استعمال کیا جا سکتا ہے۔	کی - وہ کو خیز کی ایک کیس سیسٹم لینگوچ ہے۔

جوابات

1	2	3	4	5	6	7	8	9	10
D	A	D	B	B	D	B	B	C	A

**سوال نمبر 2:** درست غلط کا انتخاب کریں۔

درست / غلط

(1) ایک آپریٹر کی قیمت زیادہ سے زیادہ 32000 ہو سکتی ہے۔

درست / غلط

(2) فارمیٹ سینٹریز ہر نشان سے شروع ہوتے ہیں۔

درست / غلط

(3) تقسیم کے آپریٹر کی ترجیح ضرب کے آپریٹر سے زیادہ ہے۔

درست / غلط

(4) (()) صارف سے ہر قسم کا ذیل ان پڑ لینے کے لیے استعمال ہوتا ہے۔

درست / غلط

(5) آئٹ پٹ آپریشن کے لیے استعمال ہوتا ہے۔

جوابات

1	2	3	4	5
غلط	درست	غلط	درست	غلط

**سوال نمبر 3:** مندرجہ ذیل کی تعریف کریں۔

صخونبر 33 پر ملاحظہ فرمائیں۔

(1) سینٹ فرینیٹر

صخونبر 29 پر ملاحظہ فرمائیں۔

(2) فارمیٹ سینٹریز

صخونبر 33 پر ملاحظہ فرمائیں۔

(3) اسکیپ سیکوننس

صخونبر 31 پر ملاحظہ فرمائیں۔

(4) scanf

صخونبر 39 پر ملاحظہ فرمائیں۔

(5) ماڈل آپریٹر

**سوال نمبر 4:** مختصر طور پر درج ذیل سوالات کے جواب دیں۔

(1) () scanf() اور (( )) getch() میں کیا فرق ہے؟

(2) یہ لیگنکون کون سا نکشن سکرین پر آئٹ پٹ ظاہر کرنے کے لیے استعمال کیا جاتا ہے؟

(3) I/O آپریٹر میں فارمیٹ سینٹریز کی وضاحت کیوں ضروری ہے؟

(4) اسکیپ سیکوننس کیا ہے؟ ہمیں ان کی ضرورت کیوں ہے؟

(5) ار تھیک آپریٹر میں کون سے آپریٹر استعمال ہوتے ہیں؟

(6) ریلیشنل آپریٹر کیا ہے؟ مثال کے ساتھ بیان کریں۔

(7) لا جیکل آپریٹر کیا ہے؟ مثال کے ساتھ بیان کریں۔

(8) یوزری آپریٹر اور باسٹری آپریٹر میں کیا فرق ہے؟

(9) == آپریٹر اور = آپریٹر میں کیا فرق ہے؟

(10) آپریٹر ز کی ترجیح سے کیا مراد ہے؟ یہ لیگنکون میں کس آپریٹر کی ترجیح سب سے زیادہ ہے؟

سوال نمبر ۵: کوڈ کے ان حصول کی آوث ہٹ کسی۔

a) #include<stdio.h>  
 void main()  
 {  
 int x = 2, y = 3, z = 6;  
 int ans1, ans2, ans3;  
 ans1 = x / z \* y;  
 ans2 = y + z / y \* 2;  
 ans3 = z / x + x \* y;  
 printf("%d %d %d", ans1, ans2, ans3);  
 }

**Output:**  
 0, 7, 9

b) #include<stdio.h>  
 void main()  
 {  
 printf("nn \n\n nnn \n\n \t ");  
 printf("nn /n/n nn/n\n");  
 }

**Output:**  
 nn  
 nnn  
 n  
 t nn /n/n nn/n

c) #include<stdio.h>  
 void main()  
 {  
 int a = 4, b;  
 float c = 2.3;  
 b = c \* a;  
 printf("%d", b);  
 }

**Output:**  
 9

d) #include<stdio.h>  
 void main()  
 {  
 int a = 4 \* 3 / ( 5 + 1 ) + 7 % 4;  
 printf("%d", a);  
 }

**Output:**  
 5

e) #include<stdio.h>  
void main()  
{  
printf("%d", (((5 > 3) && (4 > 6)) || (7 > 3)));  
}

**Output:**

1

سوال نمبر 6: کوڈ کے درج ذیل حصوں میں ایرز کی نشاندہی کریں۔

a) #include<stdio.h>  
void main()  
{  
int a, b = 13;  
b = a % 2;  
printf("Value of b is: %d, b);  
}

**Error:**

Value of variable a is not initialized.  
Inverted comma is missing after %d

b) #include<stdio.h>  
void main()  
{  
int a, b, c,  
printf("Enter First Number:");  
scanf("%d", &a);  
printf("Enter second number:");  
scanf("%d", &b);  
a + b = c;  
}

**Error:**

Semicolon is missing after int variable. Invalid syntax of a + b = c;  
The correct syntax is c=a+b;

c) #include<stdio.h>  
void main()  
{  
int num;  
printf(Enter Number:");  
scanf(%d, &num);  
};

**Error:**

Double quotes missing in printf statement and } has extra ;

#include<stdio.h>  
void main()  
{  
float f;  
printf("Enter value:");  
scanf("%c", &f);  
}

**Error:**

Square bracket is used instead of parentheses. In printf statement float variable is used with %f in scanf instead of %c.

پروگرام میں مشتمل

مشق 1: کسی کمپنی میں اجرت کا تھیں کرنے کا طریقہ درج ذیل ہے۔

Basic Salary	= Pay Rate Per Hour	X	Working Hours of Employee
Overtime Salary	= Overtime Pay Rate	X	Overtime Hours of Employee
Total Salary	= Basic Salary	+	Overtime Salary

ایک پروگرام لکھیں جس میں کام کے اوقات اور ملازم کے اور نام کئے ان پڑ کے طور پر لے جائیں۔ پروگرام میں ملازم کی کل تحریک اور حساب لگائے اور سکرین پر دکھلے کر دے۔

**Program**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int wh,oh,rate_per_hour,otrate,basic_salary,overtime_salary,total_salary;
    printf("Enter Working Hours of Employ:");
    scanf("%d",&wh);
    printf("Enter overtime working hours:");
    scanf("%d",&oh);
    printf("Enter rate per hours:");
    scanf("%d",&rate_per_hour);
    printf("Enter over Time rate per hour:");
    scanf("%d",&otrate);
    basic_salary = wh*rate_per_hour;
    overtime_salary = oh*otrate;
    total_salary = basic_salary+overtime_salary;
    printf("Total salary is = %d",total_salary);
    getch();
}
```

**Output**

```
Enter Working Hours of Employ: 20
Enter overtime working hours: 10
Enter rate per hours: 20
Enter over Time rate per hour: 10
Total Salary is = 500
```

مشن 2: ایک پروگرام کسیں جو سینٹی گریڈ میں درج حرارت کو ان پٹ کے طور پر لے اور درج حرارت کو فارن ہائیٹ میں تبدیل کرے اور آٹھ پٹ دکھائے۔

$$F = \frac{9}{5}C + 32$$

درج حرارت کو سینٹی گریڈ سے فارن ہائیٹ میں تبدیل کرنے کا فارمولے ہے۔

### Program

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    float celsius, fahrenheit;
    printf("Enter temperature in Celsius: ");
    scanf("%f", &celsius);
    //celsius to fahrenheit conversion formula
    fahrenheit = (celsius * 9 / 5) + 32;
    printf("Celsius =%.2f Farenheit=% .2f ", celsius, fahrenheit);
    getch();
}
```

### Output

```
Enter temperature in Celsius: 37
Celsius = 37.00 Farenheit = 98.60
```

مشن 3: ایک پروگرام کسیں جو ایک printf مشن کا استعمال کرتے ہوئے درج زیل آٹھ پٹ دکھائے۔

*	*	*	*
1	2	3	4

### Program

```
#include <stdio.h>
int main()
{
    printf(" *\\t *\\t *\\t *\n1\\t 2\\t 3\\t 4");
}
```

### Output

*	*	*	*
1	2	3	4

مشن 4: ایک پروگرام لکھیں جو ایک `printf` سینٹٹ کا استعمال کرتے ہوئے درج ذیل آٹھ ہٹ دکھائے۔

I am a Boy  
I live in Pakistan  
I am a Proud Pakistani

#### Program

```
#include <stdio.h>
#include <conio.h>
int main()
{
    clrscr();
    printf("I am a boy \n I live in Pakistan \n I am a Proud Pakistani");
    getch();
}
```

#### Output

I am Boy  
I live in Pakistan  
I am a Proud Pakistani

مشن 5: پیزوں کا برانڈ ہر آئینم پر 15 فیصد ڈسکاؤنٹ دیتا ہے۔ ایک خاتون اس برانڈ سے 5 شرٹس خریدتی ہے۔ ایک پروگرام لکھیں جو ڈسکاؤنٹ کے بعد کل قیمت اور خاتون کی طرف سے حاصل کردہ ڈسکاؤنٹ کی رقم کا حساب لگائے۔ شرٹ کی اصل قیمتیں یہ ہیں:

Shirt1 = 432  
Shirt2 = 320  
Shirt3 = 270  
Shirt4 = 680  
Shirt5 = 520

نوٹ: شرٹ کی قیتوں کو محفوظ کرنے کے لیے 5 خفیرات استعمال کریں۔

**Program**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    int shirt1 = 423, shirt2 = 320, shirt3 = 270, shirt4 = 680, shirt5 = 520;
    float totalprice, totaldiscount, finalprice;
    printf("First Shirt price: %d \t\t Discount is %.2f \t\t Payable price: %.2f \n", shirt1, (shirt1*0.15), (shirt1-(shirt1*0.15)));
    printf("Second Shirt price: %d \t\t Discount is %.2f \t\t Payable price: %.2f \n", shirt2, (shirt2*0.15), (shirt2-(shirt2*0.15)));
    printf("Third Shirt price: %d \t\t Discount is %.2f \t\t Payable price: %.2f \n", shirt3, (shirt3*0.15), (shirt3-(shirt3*0.15)));
    printf("Fourth Shirt price: %d \t\t Discount is %.2f \t\t Payable price: %.2f \n", shirt4, (shirt4*0.15), (shirt4-(shirt4*0.15)));
    printf("Fifth Shirt price: %d \t\t Discount is %.2f \t\t Payable price: %.2f \n", shirt5, (shirt5*0.15), (shirt5-(shirt5*0.15)));
    printf("\n\n-----\n");
    totalprice = shirt1 + shirt2 + shirt3 + shirt4 + shirt5;
    totaldiscount = (shirt1*0.15)+(shirt2*0.15)+(shirt3*0.15)+(shirt4*0.15)+(shirt5*0.15);
    finalprice = (shirt1-(shirt1*0.15))+(shirt2-(shirt2*0.15))+(shirt3-(shirt3*0.15))+(shirt4-(shirt4*0.15))+(shirt5-(shirt5*0.15));
    printf("Total price: %.2f \t Total Discount is %.2f \t Final price: %.2f \n", totalprice, totaldiscount, finalprice);
    getch();
}
```

**Output**

First Shirt Price : 423	Discount is 63.45	Payable price: 359.55
Second Shirt Price : 320	Discount is 48.00	Payable price: 272.00
Third Shirt Price : 270	Discount is 40.50	Payable price: 329.50
Fourth Shirt Price : 680	Discount is 102.00	Payable price: 578.00
Fifth Shirt Price : 520	Discount is 78.00	Payable price: 442.00

Total Price: 2213.00

Total Discount is 331.95

Final price: 1881.05

**مشن 6:** ایک پروگرام کیسے جو دو نشیب و ری لبلز کی قیتوں کا تیرے ویری ایکل کے استعمال کے بغیر تبدیل (swap) کرے۔

### Program

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    int a=10, b=20;
    printf("Before swap a=%d b=%d",a,b);
    a=a+b; //a=30 (10+20)
    b=a-b; //b=10 (30-20)
    a=a-b; //a=20 (30-10)
    printf("\nAfter swap a=%d b=%d",a,b);
    getch();
}
```

### Output

```
Before swap a = 10 b = 20
After swap a = 20 b = 10
```

**مشن 7:** ایک پروگرام کیسے جو 5 ہندسوں والا نمبر بطور ان پٹ لے اور پہلا اور آخری ہندسہ جمع کر کے جواب پر نت کرے۔

### Program

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    int number;
    printf("Enter a number with length 5 digits: ");
    scanf("%d",&number);
    printf("The sum of first and 5th digit is: %d", (number / 10000) + (number %10));
    getch();
}
```

### Output

```
Enter a number with length 5 digits: 12345
The sum of first and 5th digit is: 6
```

مشن 8: ایک پروگرام لکھیں جو صارف کی ماہانہ آمدنی اور ماہانہ اخراجات جیسے بجلی کا میل، گیس کا میل اور کھانے کے اخراجات بطور ان پڑیتے ہے۔ پروگرام ان سب کو کیلولیٹ کرے۔

- مہانہ بچت
- کل سالانہ اخراجات
- کل مہانہ اخراجات
- سالانہ اسٹریچت
- مہانہ اسٹریچت

### Program

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    int monthly_income, electricity_bill, gas_bill, food_expense, monthly_Expenses,
    monthly_savings;
    printf("Please enter your Monthly Income: ");
    scanf("%d",&monthly_income);
    printf("Please enter you Monthly Expenses: ");
    printf("\n\n\tYour Electricity Bill: ");
    scanf("%d",&electricity_bill);
    printf("\tYour Gas Bill: ");
    scanf("%d",&gas_bill);
    printf("\tYour Food Expense: ");
    scanf("%d",&food_expense);
    monthly_Expenses = electricity_bill + gas_bill + food_expense;
    printf("\n\n\t Total Monthly Expenses are: %d", monthly_Expenses);
    printf("\n\t Total Yearly Expenses are: %d", (monthly_Expenses)*12);
    monthly_savings = monthly_income - monthly_Expenses;
    printf("\n\n\t Your Monthly savings is: %d", monthly_savings);
    printf("\n\t Your Yearly savings is: %d", monthly_savings*12);
    printf("\n\n\t Average saving per month: %d", monthly_savings);
    printf("\n\t Average expense per month: %d", monthly_Expenses);
    getch();
}
```

### Output

Please enter your Monthly Income: 63000

Please enter your Monthly Expenses

Your Electricity Bill	:	2300
Your Gas Bill	:	500
Your Food Expense	:	22000
Total Monthly Expenses are	:	24800
Total Yearly Expenses are	:	297600
Your Monthly Saving is	:	38200
Your Yearly Saving is	:	458400
Average saving per month	:	38200
Average expense per month	:	24800

مشن 9: ایک پروگرام لکھیں جو ایک کریکٹر اور سیپس (steps) کی تعداد صارف سے ان پڑ لے۔ پروگرام پھر اس کریکٹر کو اتنے سیپس آگے جب کرے

Enter character: a

Enter steps: 2

Enter character: c

#### Program

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    char c;
    int steps;
    printf("Enter character: ");
    scanf("%c",&c);
    printf("\nEnter Steps: ");
    scanf("%d",&steps);
    int x = c + steps;
    printf("\nNew Character: %c",x);
    getch();
}
```

#### Output

Enter character: a

Enter steps: 2

New Character: c

مشن 10: ایک پروگرام لکھیں جو دائیے کے رادس (Radius) کو ان پڑ کے طور پر لے۔ پروگرام دائیے کا رقبہ شد کر کے سکرین پر دکھائے۔

#### Program

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    float radius;
    printf("Please enter the radius of circle: ");
    scanf(" %f ",&radius);
    printf("\n Area of circle is: %.2f ", 3.14*radius*radius);
    getch();
}
```

#### Output

Please enter the radius of circle: 20

Area of circle is: 1256.00

## سرگرمیوں کا حل

مرگری 2.1: دیے گئے کوڈی آؤٹ پٹ لکھیں۔

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("I am UPPERCASE and this is lowercase");
    getch();
}
```

**Output**

I am UPPERCASE but this is lowercase

مرگری 2.2: سکرین پر اپنے نام کا پہلا حصہ اپر کیس (upper case) اور دوسرا حصہ لوئر کیس (lower case) میں دکھانے کے لیے پروگرام لکھیں۔

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    printf("MUBASHIR hussain");
    getch();
}
```

**Output**

MUBASHIR hussain

برگری 2.3: ایک پروگرام لکھیں جو صارف سے رو لنگر، نمبر اس کی پر شیخ اور گریڈ بلور ان پڑ لے اور اس طرح سے فارمیٹ کی ہوئی آئٹ پٹ دکھائے۔

Roll no	:	input value
Percentage	:	input value %
Grade	:	input value

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    int roll_number;
    float percentage;
    char grade;
    printf("Please enter your Roll Number: ");
    scanf("%d",&roll_number);
    printf("Please enter your Percentage: ");
    scanf("%f ",&percentage);
    printf("Please enter your Grade: ");
    scanf(" %c", &grade); // All use SPACE before %c like " %c" not like " %c ";
    printf("Roll no \t\t: \t\t%d \n",roll_number);
    printf("Percentage \t\t: \t\t%f \n",percentage);
    printf("Grade \t\t\t: \t\t%c",grade);
    getch();
}
```

#### Output

Please enter your Roll Number :	36
Please enter your Percentage :	89
Please enter your Grade :	A
Roll no :	36
Percentage :	89.000000
Grade :	A

مرگری 4.2: ایک پروگرام لکھیں جو چکر (Square) کے ایک طرف کی لمبائی ان پڑتے ہے اور اس کے رقبے کا حساب لگاتے۔

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    int n;
    printf("Enter any number: ");
    scanf("%d",&n);
    printf("\nArea of Square =%d",n*n);
    getch();
    return 0;
    getch();
}
```

#### Output

Enter any number: 6  
Area of Square= 36

مرگری 4.5: ایک پروگرام لکھیں جو جار A اور B میں موجود گیندوں کی تعداد ان پڑتے ہے اور گیندوں کی کل تعداد سکرین پر دکھاتے۔

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    int a,b,sum=0;
    printf("Enter number of balls in Jar A:");
    scanf("%d",&a);
    printf("Enter number of balls in Jar B:");
    scanf("%d",&b);
    sum=a+b;
    printf("Total Number of Balls = %d",sum);
    getch();
    return 0;
}
```

#### Output

Enter number of balls in Jar A: 10  
Enter number of balls in Jar B: 15  
Total Number of Balls = 25

مرگری 2.6: ایک پروگرام لکھیں جو شرت کی اصل قیمت اور ڈسکاؤنٹ پر سنتج صارف سے ان پڑ لے اور شرت کی اصل قیمت، قیمت پر کیا گیا ڈسکاؤنٹ اور ڈسکاؤنٹ کے بعد والی قیمت سکرین پر دکھائے۔

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    int op,dp;
    printf("Enter Original Price of shirt ");
    scanf("%d",&op);
    printf("Enter Discount in percentage");
    scanf("%d",&dp);
    printf("\nOriginal Price of shirt = %d",op);
    printf("\nDiscount on Shirt = %d",dp*op/100);
    printf("\nPrice of shirt after discount = %d",op-dp);
    getch();
    return 0;
}
```

**Output**

Enter Original Price of shirt 290
Enter Discount in percentage 10
Original Price of shirt= 290
Discount on Shirt= 29
Price of shirt after discount= 261

مرگری 2.7: ایک پروگرام لکھیں جو صارف سے 2 ہندسوں والا نمبر ان پڑ لے اور ایک ہندسے کو دوسرے ہندسے سے ضرب دے کر آٹھ پڑ دکھائے۔

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    int digit1, digit2;
    printf("Please enter the first digit number: ");
    scanf("%d",&digit1);
    printf("Please enter the second digit number: ");
    scanf("%d",&digit2);
    printf("\nProduct of first and second digit is: %d", digit1*digit2);
    getch();
}
```

**Output**

Please enter the first digit number: 6
Please enter the second digit number: 5
Product of first and second digit is: 30

**مرگی 8.2:** ایک پروگرام لکھیں جو سینٹ انپٹ لے اور ان کے برابر گئے، منٹ اور سینٹ شادر کے۔

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    float seconds, minutes, hours;
    printf("Please enter the seconds: ");
    scanf(" %f ",&seconds);
    minutes = seconds / 60;
    hours = minutes / 60;
    printf("\nTotal Seconds: %.2f\nTotal Minutes: %.2f\nTotal Hours: %.2f ",seconds,minutes,hours);
    getch();
}
```

### Output

Please enter the seconds: 234

Total Seconds: 234.00

Total Minutes: 3.90

Total Hours: 0.07

**مرگی 9.2:** الجبرا کے ایک پر بیٹھ کو C ایک پر بیٹھ میں تبدیل کریں۔

$$x = 6y + x$$

$$x = yz^3 + 3y$$

$$z = x + \frac{y^2}{3x}$$

$$z = (x - 2)^2 + 3y$$

$$y = \left( x + \frac{3z}{2} \right) + z^3 + \frac{x}{z}$$

### Solution:

$$x = 6 * y + z$$

$$x = y * z * z * z + 3 * y$$

$$z = x + y * y / 3 * x$$

$$z = (x-2) * (x-2) + 3 * y$$

$$y = (x + 3 * z / 2) + z * z * z + x / z$$

سرگری 2.10: پیچ دی گئی ایک پریشن کا بولین رزلت بتائیں۔ جبکہ ویری لبلز اور ان کی قیمتیں ہیں۔  $y=7, x=3$

$$(2 + 5) > y$$

$$(x+4) == y$$

$$x != (y - 4)$$

$$(y / 2) >= x$$

$$-1 < x$$

$$(x * 3) <= 20$$

**Solution:**

$$(2 + 5) > y$$

False

$$(x+4) == y$$

True

$$x != (y - 4)$$

False

$$(y / 2) >= x$$

True

$$-1 < x$$

True

$$(x * 3) <= 20$$

True

سرگری 2.11: فرض کریں دیے گئے ویری لبلز کی قیمتیں ہیں  $x=4, y=7, z=8$  جو ان ایک پریشن بتائیں۔

$$x == 2 \text{ || } y == 8$$

$$7 >= y \&\& z < 5$$

$$z >= 5 \text{ || } x <= -3$$

$$y == 7 \&\& !(true)$$

$$x != y \text{ || } y < 5$$

$$!(z > x)$$

**Solution:**

$$x == 2 \text{ || } y == 8$$

False

$$7 >= y \&\& z < 5$$

True

$$z >= 5 \text{ || } x <= -3$$

True

$$y == 7 \&\& !(true)$$

False

$$x != y \text{ || } y < 5$$

True

سرگری 2.12: ان ایک پریشن کے جوابات لکھیں۔

$$16 / (5 + 3)$$

$$7 + 3 * (12 + 2)$$

$$25 \% 3 * 4$$

$$34 - 9 * 2 / (3 * 3)$$

$$18 / (15 - 3 * 2)$$

**Solution:**

$$16 / (5 + 3)$$

= 2

$$7 + 3 * (12 + 2)$$

= 49

$$25 \% 3 * 4$$

= 4

$$34 - 9 * 2 / (3 * 3)$$

= 32

$$18 / (15 - 3 * 2)$$

= 2

# یوزر انٹریکشن

61

## کشمیر الاحتجاجی سوالات

D	C	B	A	سوالات	نمبر
تمام	موباکل	پرمنٹ	کپیوٹر	----- ایک ایسا آئہ ہے جو ڈینا کو ان پٹ کے طور پر لیتا ہے۔ ڈینا کو پڑ دیں کرتا ہے اور آٹھ پٹ دیتا ہے۔	1
اور بدوں la	لینگوچ	معاری لائبریری	کی - ورڈ	ہر پروگرام میں لینگوچ کی اپنی ----- اور ----- ہوتے ہیں جو I/O آپریٹر سر انجام دیتے ہیں۔	2
تمام	آٹھ پٹ	ان پٹ	پروگرام	printf کریں پر ----- دکھانے کے لیے ہی پروگرام میں لینگوچ میں ایک بلٹ ان (built-in) فناش ہے۔	3
clrscr	printf	scanf	getch	----- فناش صارف سے ان پٹ حاصل کرنے کے لیے استعمال ہوتا ہے۔	4
پرنس فارمینڈ	پرنسیپ سیکونس	غیر پرنس فارمینڈ	اسکیپ سیکونس	printf کا نام ----- سے لکھا ہے جو فارمیٹ شدہ آٹھ پٹ کو سکرین پر پرنس کرنے کے لیے استعمال ہوتا ہے۔	5
تمام	char	float	int	فارمیٹ پسیغائر ۳ ہزار ڈینا کی قسم ----- دکھاتا ہے۔	6
تمام	char	float	int	فارمیٹ پسیغائر ۵ ہزار ڈینا کی قسم ----- دکھاتا ہے۔	7
8	7	6	5	جب ہم فلوٹ دیجیٹو ظاہر کرنے کے لیے ۴% کا استعمال کرتے ہیں تو یہ اعشاریہ کے بعد ----- ہندسے دکھاتا ہے۔	8
گائیڈ لائن	ہدایات	آٹھ پٹ	ان پٹ	سی لینگوچ میں ایک بلٹ ان فناش ہے جو صارف سے دیری ایڈمز میں ----- لیتا ہے۔	9
منطقی آپریٹر	فارمیٹ پسیغائر	اسکیپ سیکونس	scanf	فناش میں ہم ----- کی مدد سے ان پٹ ڈینا کی متوقع قسم کی وضاحت کرتے ہیں۔	10
5	4	3	2	----- فناش کے اہم حصے -----	11
ٹرپل داوین کے اندر	ڈیز می بریکٹ کے اندر	ڈبل داوین کے اندر	سنکل داوین کے اندر	فناش کا پہلا حصہ ----- ہے۔	12

## یوزر انٹریکشن

62

ان پٹ	اچر کیوٹ	کپاٹر	ہدایات	نکشن & اور بڑے بغیر پروگرام ۔۔۔۔۔ تو ہو جاتا ہے scanf مگر موقع متعارف نہیں دیتا ہے۔	13
s	.	%	&	نکشن میں ۔۔۔۔۔ اور بڑے بغیر کو بھول جانا بہت عام سی غلطی scanf ہے۔	14
clrscr	scanf	getch()	printf	صرف سے ایک کریکٹر لینے کے لیے _____ نکشن استعمال کر جاتا ہے۔	15
iostream.h	conio.h	math.h	stdio.h	نکشن استعمال کرنے کے لیے ہمیں لا جبری getch() کو پروگرام کے بعد ریکشن میں شامل کرنے کی ضرورت پڑتی ہے۔	16
ctrl	shift	esc	Enter	جب ہم scanf کے ذریعے کریکٹر پڑھتے ہیں تو ہمیں پروگرام کو آگے چلانے کے لیے ۔۔۔۔۔ دبائے کی ضرورت ہوتی ہے۔	17
حقیقی آپریٹر	شیئٹ فریزیٹر	اسکیپ سیکونس	فارمیٹ پیسٹنائزر	کپاٹر کے لیے شاخت کندہ ہے جو لائئن کے اختتام کی نشاندہی کرتا ہے۔	18
.	%	;	!	سی لیکچون میں _____ بطور شیئٹ فریزیٹر استعمال ہوتا ہے۔	19
فارمیٹ پیسٹنائزر	اسکیپ سیکونس	شیئٹ فریزیٹر	فریزیٹر	نکشن میں دوین ("") کے اندر استعمال printf _____ ہوتے ہیں۔	20
آپریٹر	فارمیٹ	فارمیٹ پیسٹنائزر	اسکیپ سیکونس	printf("My name is \\"Ali\\\""); مندرجہ بالا مثال میں " " ایک _____ ہے۔	21
تمام	شیئٹ فریزیٹر	اسکیپ سیکونس	فارمیٹ پیسٹنائزر	دو حروف پر مشتمل ہے۔	22
ستکل کوٹ سکرین پر دکھانا	چھپا کریکٹر مٹا	الٹ کی آواز دیتا	بیک سلیش سکرین پر دکھانا	اسکیپ سیکونس () کا معنادہ کیا ہے؟	23

بک سلیش سکرین پر دکھانا	الرٹ کی آواز دینا	چھلا کر یکٹر منانا	سنگل کوٹ سکرین پر دکھانا	اسکیپ سیکونس a کا مقصد کیا ہے؟	24
\a	\b	\n	\t	اسکیپ کر یکٹر کے بعد _____ واضح کرتا ہے کہ کمر (cursor) کو اگلی لائن کے شروع پر لے کر جاتا ہے۔	25
\n	\c	\b	\a	آؤٹ پٹ کو متعدد لائنوں پر پرمنٹ کرنے کے لیے کون سا اسکیپ سیکونس استعمال کیا جاتا ہے؟	26
\l	\t	\n	\m	اسکیپ سیکونس I/O نقش کو بناتا ہے کہ افتن طور پر اگلے تیپ سٹاپ (Tab Stop) پر جاتا ہے۔	27
تمام	ریلیشل آپریٹر	ارٹھمیٹک آپریٹر	اسائنسٹ آپریٹر	آپریٹر کی اقسام _____ ہیں۔	28
ریلیشل آپریٹر	منطقی آپریٹر	اسائنسٹ آپریٹر	ارٹھمیٹک آپریٹر	ایک متغیر کو قیمت تفویض کرنے کے لیے استعمال کیا جاتا ہے یا ایک متغیر کی قیمت کو دوسرے متغیر میں رکھنے کے لیے استعمال ہوتا ہے۔	29
%	!	\$	=	کی میں _____ اسائنسٹ آپریٹر کے طور پر استعمال ہوتا ہے۔	30
ریلیشل آپریٹر	منطقی آپریٹر	اسائنسٹ آپریٹر	ارٹھمیٹک آپریٹر	ڈیاپر ریاضی کے کام کرنے کے لیے استعمال ہوتے ہیں۔	31
جمع آپریٹر	تقسیم آپریٹر	تفريق آپریٹر	ضرب آپریٹر	/ آپریٹر _____ کا نام کیا ہے؟	32
ماڈولس آپریٹر	جمع آپریٹر	ضرب آپریٹر	تقسیم آپریٹر	بائیں اوپر بیڈ کی قیمت کو دائیں اوپر بیڈ کی قیمت پر تقسیم کرتا ہے۔	33
تمام	int	var	char	اگر دونوں اوپر بیڈز int تابپ ہوں تو تقسیم کا نتیجہ کبھی _____ تابپ کا ہو گا۔	34

تفہیم آپریٹر	ضرب آپریٹر	ارٹھمیٹک آپریٹر	منطقی آپریٹر	ایک بائزی آپریٹر ہے جو دو نمبروں کو ضرب دتتا ہے۔	35
منطقی آپریٹر	تفہیم آپریٹر	ضرب آپریٹر	ارٹھمیٹک آپریٹر	کا آپریٹر (+) دو اور یونڈ کو جمع کرتا ہے۔	36
-1	1.8	1.5	1	شینٹ - - - a یا a-- ویری ایتل a کی قیمت میں کوکم کرنے کے لیے استعمال کیا جاتا ہے۔	37
تفریق آپریٹر	تفہیم آپریٹر	ضرب آپریٹر	ارٹھمیٹک آپریٹر	بائیک اور یونڈ سے دو بائیک اور یونڈ کو تفریق کرتا ہے۔	38
تفریق آپریٹر	تفہیم آپریٹر	ماڈولس آپریٹر	ضرب آپریٹر	% کا نام ہے۔	39
ارٹھمیٹک آپریٹر	تفریق آپریٹر	تفہیم آپریٹر	ماڈولس آپریٹر	بائیک اور یونڈ کو دو بائیک اور یونڈ پر تقسیم کرتا ہے اور تقسیم کے بعد بچنے والی باقی رقم لوٹاتا ہے۔	40
بائزی آپریٹر	منطقی آپریٹر	ریلیشل آپریٹر	ارٹھمیٹک آپریٹر	دو قیتوں کے درمیان تعلق کا تعین کرنے کے لیے اتنا موافق کرتے ہیں۔	41
C اور C دونوں	char	Numeric	int	C لینگوچ میں اور تابکے ذیل پر ریلیشل آپریٹر زانجمن دینے کی اجازت دیتی ہے۔	42
3	2	1	0	ریلیشل آپریٹر میں درست دلیل کی نمائندگی --- سے ہوتی ہے۔	43
3	2	1	0	ریلیشل آپریٹر میں مخلوط دلیل کی نمائندگی --- سے ہوتی ہے۔	44
/	+	=	==	سی لینگوچ میں دو ایکپریشن کی مساوات کی جانچ کے لیے استعمال ہوتا ہے کہ وہ برابر ہیں کہ نہیں۔	45
%	*	=	==	آپریٹر دو بائیک ایکپریشن کو بائیک دالے ویری ایتل سے منسوب کرتا ہے۔	46
ریلیشل آپریٹر	ارٹھمیٹک آپریٹر	اسائنس آپریٹر	منطقی آپریٹر	بولین ایکپریشن پر آپریشن سرانجام دیتے ہیں اور اس کے نتیجے میں جواب بھی بولین ایکپریشن ہوتا ہے۔	47

## یوزر انٹریکشن

65

NOT	NAND	OR	AND	ان میں سے کن سامنگی آپریٹر نہیں ہے۔	48
NOT	NAND	OR	AND	دو یوں ایک پریشن کو بطور اور بیند لیتا ہے اور اگر اس کے دونوں ایک پریشن درست (true) ہوں تو نتیجہ درست آتا ہے۔	49
NOT	NAND	OR	AND	دو یوں ایک پریشن لیتا ہے اور کم از کم ایک ایک پریشن درست (true) ہونے پر جواب درست دیتا ہے۔	50
NAND	OR	AND	NOT	بولین ایک پریشن کی قیمت کی نفی یا الٹ کرتا ہے۔ یہ اسے درست (true) بنادتا ہے اگر یہ غلط ہے اور اگر یہ غلط (false) ہے تو اسے درست بنادتا ہے۔	51
منطقی آپریٹر	بائزی آپریٹر	یوزری آپریٹر	ریلیشنل آپریٹر	کو آپریٹ کرنے کے لیے دو ایک پریشن کی ضرورت ہوتی ہے۔	52
ریلیشنل آپریٹر	منطقی آپریٹر	بائزی آپریٹر	یوزری آپریٹر	صرف ایک آپریٹ پر لا گو ہوتے ہیں۔	53
!	()	^	&	کون سے آپریٹ کی ترجیح زیادہ ہے؟	54
منطقی	بائزی	یوزری	ترزی	آپریٹر میں اور بیند پر لا گو ہوتے ہیں۔	55
clrscr();	printf	scanf	getch();	فتاش سی لینگوچ ایڈیٹر کی آئٹ پٹ سکرین کو صاف کرنے کے لیے استعمال کیا جاتا ہے۔	56

جوابات

1	2	3	4	5	6	7	8	9	10
A	D	C	B	D	B	C	B	A	C
11	12	13	14	15	16	17	18	19	20
A	B	C	A	B	C	A	C	B	C
21	22	23	24	25	26	27	28	29	30
A	B	A	C	B	D	C	D	B	A
31	32	33	34	35	36	37	38	39	40
A	C	A	C	C	A	A	D	B	A
41	42	43	44	45	46	47	48	49	50
B	D	B	A	A	B	A	C	A	B
51	52	53	54	55	56				
A	C	A	C	A	D				

## مختصر سوالات

1: کپیور کی وضاحت کریں۔

جواب: کپیور ایک ایسا آن پٹ کے طور پر لیتا ہے جو ڈائیا کوپر سس کرتا ہے اور آئٹ پٹ دیتا ہے۔

2: printf فونکشن کی وضاحت کریں؟

جواب: printf سکرین پر آئٹ پٹ دکھانے کے لیے سی پروگرامنگ لینگوچ میں ایک بلت ان (built-in) فونکشن ہے۔ اس کا نام "پرنٹ فارمیٹ" (print formatted) ہے جو فارمیٹ شدہ آئٹ پٹ کو سکرین پر پرنٹ کرنے کے لیے استعمال ہوتا ہے۔ تمام ڈیا کی اقسام printf فونکشن کے ساتھ خالہ کی جاسکتی ہیں۔

3: فارمیٹ پیسٹھار کیا ہے؟

جواب: فارمیٹ پیسٹھار ڈیا کسی فیلڈ کی چوڑائی اور سکرین پر ظاہر ہونے والے متغیر کی ولیوں کی ملک کی نمائندگی کرتا ہے۔ ایک فارمیٹ پیسٹھار ایک ایسی علامت ہے کہ سے شروع ہوتا ہے۔ فارمیٹ پیسٹھار ان پٹ اور آئٹ پٹ دونوں مشینٹ کے لیے استعمال ہوتا ہے۔

4: C لینگوچ میں مختلف ڈیا کی اقسام میں استعمال ہونے والے فارمیٹ پیسٹھار لکھیں؟

جواب: مندرجہ ذیل نیچل سی لینگوچ میں مختلف ڈیا کس کے مقابلے میں فارمیٹ پیسٹھار دکھاتا ہے۔

ڈیا کا نام	فارمیٹ پیسٹھار
int	% d or % i
float	% f
char	% c

5: اگر int age = 35; , float height= 5.8; ہو تو اس مشینٹ کی آئٹ پٹ کیا ہو گی؟

جواب:

printf("My age is %d and my height is %. 2f", age, height);

My age is 35 and my height is 5.80

6: اس کوڈ کی آئٹ پٹ کیا ہو گی؟

```
#include<stdio.h>
void main()
{
printf("Sum of 23 and 45 is %d", 23 + 45);
}
```

Sum of 23 and 45 is 68

جواب:

7: () scanf کا مقصد کیا ہے؟

جواب: scanf سی لینگوچ میں ایک بلٹ ان فنکشن ہے جو صارف سے دیری ایبلز میں ان پٹ لیتا ہے۔ فنکشن میں ہم فارمیٹ سپیسیفارر کی مدد سے ان پٹ ڈیٹا کی موقع قسم کی دھاخت کرتے ہیں۔ اگر صارف اشیجروٹ ناٹاپ میں ڈیٹا دیتا ہے تو scanf میں بیان کردہ فارمیٹ سپیسیفارر % یا % ہونا چاہیے۔

8: scanf میں عام غلطی کیا ہے؟

جواب: scanf فنکشن میں & اور پریز کو بھول جانا بہت عام سی غلطی ہے۔ اور پریز کے بغیر پر گرام ایگزیکیوٹ تو جاتا ہے مگر موقع نتائج نہیں دیتا ہے۔

9: () getch کا کیا استعمال ہے؟

جواب: () getch فنکشن صارف سے ایک کریکٹر لینے کے لیے استعمال کیا جاتا ہے۔ صارف کا داخل کر دہ کریکٹر سکرین پر ظاہر نہیں ہوتا ہے۔ یہ فنکشن عام طور پر پر گرام کی ایگزیکیوٹ کو روکنے کے لیے استعمال کیا جاتا ہے کیونکہ پر گرام اس وقت تک آگے نہیں چلتا جب تک کہ صارف کوئی بٹن نہ دیا کے۔

10: کس لامبرری فنکشن کو () getch شامل کرنے کے لیے استعمال کیا جاتا ہے؟

جواب: () getch فنکشن کو استعمال کرنے کے لیے ہمیں لامبرری conio.h کو پر گرام کے بیڈر سیکشن میں شامل کرنے کی ضرورت پڑتی ہے۔

11: شیفت فرمینیٹ کی تعریف کریں؟

جواب: شیفت فرمینیٹ کپائلر کے لیے شاخت کندہ ہے جو لائن کے اختتام کی نشاندہی کرتا ہے۔ سی لینگوچ میں سی کو ان (;) بطور شیفت فرمینیٹ استعمال ہوتا ہے۔ اگر ہر شیفت کے آخر میں سی کو لنڈنگس تو ایر (Error) آ جاتا ہے۔

12: سی لینگوچ میں اسکیپ سیکونس کا کیا مقصد ہے؟

جواب: اسکیپ سیکونس printf فنکشن میں واوین ("") کے اندر استعمال ہوتے ہیں۔ جو printf کو استعمال کرتے ہوئے آؤٹ پٹ دکھانے کے اپنے طریقہ کار کو تبدیل کرنے پر مجبور کرتے ہیں۔

13: سی لینگوچ اسکیپ سیکونس کو مثال سے بیان کریں۔

جواب:

My name is "Ali"

مندرج بالا شیفت کی آؤٹ پٹ یہ ہے۔

مندرج بالا مثال میں " " ایک اسکیپ سیکونس ہے۔ یہ printf کو "کپیوٹر اسکرین پر" ظاہر کرنے کا سبب ہتا ہے۔

14: اسکیپ سیکونس کی تکمیل بیان کریں؟

جواب: اسکیپ سیکونس دو حروف پر مشتمل ہے۔ پہلا کریکٹر ہمیشہ یک سلیش (back slash) (\) ہوتا ہے اور دوسرا کریکٹر مطلوب فنکشنیٹی کے مطابق آتا ہے جسے ہم حاصل کرنا چاہتے ہیں۔ یہک سلیش ( \ ) کو اسکیپ کر کریکٹر کہا جاتا ہے جو اسکیپ کے بارے میں نشاندہی کرنے کے لیے ہر اسکیپ سیکونس سے منسلک ہوتا ہے۔ اسکیپ کر کریکٹر اور اس کے بعد آئنے والا کر کریکٹر سکرین پر ظاہر نہیں ہوتے ہیں لیکن یہ مخصوص کام کرتے ہیں جو ان سے منسوب کیا گیا ہے۔

15: n کا مقصد کیا ہے؟

جواب: اسکیپ کر کریکٹر کے بعد n واضح کرتا ہے کہ کرسر (cursor) کو الگی لائن (New Line) کے شروع پر لے کر جاتا ہے۔ یہ اسکیپ سیکونس ایک سے زیادہ لائنوں پر آؤٹ پٹ پر نٹ کرنے کے لیے استعمال ہوتا ہے۔

16: ٹیب (Tab) کا مقصد بیان کریں؟

جواب: اسکیپ سیکوئنس \t، ۱/۰ لائن کو بتاتا ہے کہ افتن طور پر اگلے ٹیب ستاپ (Tab Stop) پر جاتا ہے۔ ایک ٹیب ستاپ 8 چیکس (spaces) کا مجموعہ ہوتا ہے۔ \t کا استعمال کر سر کو اگلے ٹیب ستاپ پر لے جاتا ہے۔

17: آپریٹر کی تعریف کریں؟

جواب: آپریٹر زریاضی کی علاویں ہیں جو اپرینڈ (operator) پر کچھ آپریشن کرتی ہیں۔ اور یہ دوسری ای بلزیاڈیوز ہو سکتی ہیں۔

18: جنیادی آپریٹر کے نام لکھیں؟

جواب: C میں بہت سے آپریٹر استعمال ہوتے ہیں ان میں سے کچھ یہ ہیں:

- |                        |   |
|------------------------|---|
| (Logical operators)    | • مختلط آپریٹر (Assignment operator)    |
| (Relational operators) | • ریلیشنل آپریٹر (Arithmetic operators) |

19: اسائنسٹ آپریٹر کیا ہے؟

جواب: اسائنسٹ آپریٹر ایک متغیر کو قیمت تفویض کرنے کے لیے استعمال کیا جاتا ہے یا ایک متغیر کی قیمت کو دوسرے متغیر میں رکھنے کے لیے استعمال ہوتا ہے۔ بر ابری کی علامت (=) کوئی لیٹکوئچ میں اسائنسٹ آپریٹر کے طور پر استعمال کیا جاتا ہے۔ مثال:

20: ایک پروگرام لکھیں جو دو ای بلزیاڈی ای بلزی کی قیتوں کا آپس میں تہادل (swap) کرے۔

```
#include<stdio.h>
void main()
{
    int a = 2, b = 3, temp;
    temp = a;
    a = b;
    b = temp;
    printf("Value of b after swapping: %d\n", b);
    printf("Value of a after swapping: %d\n", a);
}
```

21: ار تمیلک آپریٹر کا استعمال کیا ہے؟

جواب: ار تمیلک آپریٹر زریاضی کے کام کرنے کے لیے استعمال ہوتے ہیں۔

22: قسم آپریٹر کیا ہے؟

جواب: تقسیم کا آپریٹر (/) بائیں اور یونڈ کی قیمت کو دائیں اور یونڈ کی قیمت پر تقسیم کرتا ہے۔ جیسے:

`float result = 3.0 / 2.0;`

یونڈ پلے کے بعد متغیر کا نتیجہ 1.5 آتا ہے۔

23: ضرب آپریٹر کیا ہے؟

جواب: ضرب آپریٹر ( $*$ ) ایک بائرنی آپریٹر ہے جو دو نمبروں کو ضرب دیتا ہے۔ جیسے: اس سینٹنٹ پر عمل درآمد کے بعد ویری ایبل (multiply) کی قیمت 25 ہو جائے گی۔

24: جمع کا آپریٹر کیا ہے؟

جواب: جمع کا آپریٹر ( $+$ ) دو اپریٹڈ کو جمع کرتا ہے۔ جیسے: متغیر add کی قیمت 20 ہو جائے گی۔

25: سینٹنٹ  $a = a + 1$  اور  $C \text{ لینگوچ میں کیوں استعمال کیا جاتا ہے؟}$

جواب: سینٹنٹ  $a = a + 1$  ویری ایبل a کی قیمت میں 1 کا اضافہ کرنے کے لیے استعمال ہوتی ہے۔ C لینگوچ میں اس سینٹنٹ کو;  $a = a + 1$  یا  $a++$  بھی لکھا جاسکتا ہے۔ اسی طرح;  $-a$  یا  $a -$  ویری ایبل a کی قیمت میں 1 کو کم کرنے کے لیے استعمال کیا جاتا ہے۔

26: تفریق آپریٹر کیا ہے؟

جواب: تفریق کا آپریٹر ( $-$ ) باس اپریٹڈ سے دائیں اپریٹڈ کو تفریق کرتا ہے۔ جیسے: تفریق کرنے کے بعد متغیر result کی قیمت 5 ہو جاتی ہے۔

27: ماڈولس آپریٹر کیا ہے؟

جواب: ماڈولس آپریٹر ( $\%$ ) باس اپریٹڈ کو داںکیں اپریٹڈ پر تقسیم کرتا ہے اور تھیس کے بعد بچنے والی باقی رقم لوٹاتا ہے۔ ماڈولس آپریٹر اٹھیجہر ڈیٹا ناپ پر کام کرتا ہے۔ جیسے:

اگر  $b = 14$  کو 3 سے تقسیم کریں تو 2 باقی بچے گا۔ اس لیے ویری ایبل remaining کی قیمت 2 ہو گی۔

28: C لینگوچ میں ار تمیک سینٹنٹ لکھتے ہوئے عام غلطی کیا کی جاتی ہے؟

جواب: اگرچہ C لینگوچ میں ار تمیک سینٹنٹ لکھتے ہوئے ایک عام غلطی الجبرا کے روزمرہ کے تو انہیں کو استعمال کرنا ہے۔ مثلاً  $y = 6^x$  کو  $y = x^3$  کو  $x = 0$  دیجئے۔ اس کے نتیجے میں کپاکلر ایرر (Compiler Error) آتا ہے۔

29: بریلیشل آپریٹر زدی لینگوچ میں کیوں استعمال ہوتے ہیں؟

جواب: بریلیشل آپریٹر زدی دو قیتوں کے درمیان تعلق کالین کرنے کے لیے انکا موازنہ کرتے ہیں۔ بریلیشل آپریٹر زدی خاخت کرتے ہیں کہ قیتوں برابر ہیں یا برابر نہیں۔ ایک قیمت دوسری قیمت سے بڑی ہے یا چھوٹی ہے۔ سی لینگوچ میں نیو میر ک اور char ناپ کے ڈینا پر بریلیشل آپریٹر زدی انجام دینے کی اجازت دیتی ہے۔

30: اسائٹنٹ آپریٹر (=) اور برابر کے آپریٹر (==) میں کیا فرق ہے؟

جواب: سی لینگوچ میں (==) آپریٹر دو ایک پریشن کی مساوات کی جانچ کے لیے استعمال ہوتا ہے کہ وہ برابر ہیں کہ نہیں۔ جبکہ (=) آپریٹر دائیں والی ایک پریشن کو باسیں والے ویری ایبل سے منسوب کرتا ہے۔ ڈبل ایکوکل (equal) آپریٹر (==) چیک کرتا ہے کہ دونوں طرف یہ آپریٹر زدی برابر ہیں یا نہیں۔ سنگل ایکوکل آپریٹر (=) دائیں اپریٹڈ زدی کو باسیں اپریٹڈ سے منسوب کرتا ہے۔

31: مطلق آپریٹر اور ارجمنیک آپریٹر کت در میان فرق بیان کریں؟

ارجمنیک آپریٹر	مطلق آپریٹر
ارجمنیک آپریٹر زندگی پر ریاضی کے کام کرنے کے لیے استعمال ہوتے ہیں۔ تقسیم، ضرب، جمع، تفریق اور ماڈولس ارجمنیک آپریٹر کی حمیں ہیں۔	مطلق آپریٹر زبولین ایک پریشن پر آپریشن سرانجام دیتے ہیں اور اس کے نتیجے میں جواب بھی بولین ایک پریشن ہوتا ہے۔ ریلیشنل آپریشن کا جواب بولین ایک پریشن ہے۔ ایک سے زیادہ ریلیشنل ایک پریشن کا جائزہ لینے کے لیے مطلق آپریٹر کو سرانجام دیا جاسکتا ہے۔

32: تقسیم آپریٹر اور ضرب آپریٹر میں کیا فرق ہے؟

ضرب آپریٹر	تقسیم آپریٹر
ضرب آپریٹر ( $*$ ) ایک بائرنی آپریٹر ہے جو دونہردوں کو ضرب دیتا ہے۔ چیزیں: <code>int multiply = 5 * 5;</code> اس سینکڑ پر عمل درآمد کے بعد ویری اینبل (multiply) کی قیمت 25 ہو گی۔	تقسیم کا آپریٹر (/) بائس اور پرینڈ کی قیمت کو دو ایکس اور پرینڈ کی قیمت پر تقسیم کرتا ہے۔ چیزیں: <code>float result = 3.0 / 2.0;</code> سینکڑ پلے کے بعد متغیر کا نتیجہ 1.5 آتا ہے۔

33: مطلق آپریٹر کی وضاحت کریں؟

جواب: مطلق آپریٹر زبولین ایک پریشن پر آپریشن سرانجام دیتے ہیں اور اس کے نتیجے میں جواب بھی بولین ایک پریشن ہوتا ہے۔ ریلیشنل آپریشن کا جواب بولین ایک پریشن ہے۔ ایک سے زیادہ ریلیشنل ایک پریشن کا جائزہ لینے کے لیے مطلق آپریٹر کو سرانجام دیا جاسکتا ہے۔

34: مطلق آپریٹر کا نام لکھیں؟

جواب: مندرجہ ذیل بیانیاتی مطلق آپریٹر اور ان کی تفصیل کو ظاہر کرتا ہے۔

تفصیل	آپریٹر
Logical AND	<code>&amp;&amp;</code>
Logical OR	<code>  </code>
Logical NOT	<code>!</code>

**آپریٹر AND :35** کیا ہے؟ اس کو اڑو تھہ نیل کی مدد سے بیان کریں؟

جواب: AND آپریٹر (**&&**) دو بولین ایکپریشن کو بطور اور پڑنے لیتا ہے اور اگر اس کے دونوں ایکپریشن درست (true) ہوں تو نتیجہ درست آتا ہے۔ اگر کوئی بھی آیا ایکپریشن غلط (false) ہو تو جواب غلط آتا ہے۔ AND آپریٹر کے لیے اڑو تھہ نیل ذیل میں دکھایا گیا ہے۔

رزک	ایکپریشن 2	آپریٹر	ایکپریشن 1
False	False	<b>&amp;&amp;</b>	False
False	True	<b>&amp;&amp;</b>	False
False	False	<b>&amp;&amp;</b>	True
True	True	<b>&amp;&amp;</b>	True

**آپریٹر OR :36** کیا ہے؟

جواب: OR آپریٹر دو بولین ایکپریشن لیتا ہے اور کم از کم ایک ایکپریشن درست (true) ہونے پر جواب درست دیتا ہے۔

**OR :37** آپریٹر کے لیے اڑو تھہ نیل بنائیں؟

جواب: OR آپریٹر کے لیے اڑو تھہ نیل ذیل میں دکھایا گیا ہے:

رزک	ایکپریشن 2	آپریٹر	ایکپریشن 1
False	False	<b>  </b>	False
True	True	<b>  </b>	False
True	False	<b>  </b>	True
True	True	<b>  </b>	True

38: NOT آپریٹر (!) کیا ہے؟ ٹرو جھٹ نیل بھی بنائیں؟

جواب: NOT آپریٹر بولیں ایک پریشن کی قیمت کی فنی یا الٹ کرتا ہے۔ یہ اسے درست (true) بنادتا ہے اگر یہ غلط ہے اور اگر یہ غلط (false) ہے تو اسے درست بنادتا ہے۔ آپریٹر کے لیے ٹرو جھٹ نیل ذیل میں دیا گیا ہے۔

رزک	آپریٹر	ایک پریشن
True	!	False
False	!	True

39: یو زری اور بائسٹری آپریٹرز میں فرق کریں۔

بائسٹری آپریٹر	یو زری آپریٹر
بائسٹری آپریٹرز کو آپریشن کرنے کے لیے دو ایک پریشن کی ضرورت ہوتی ہے۔ مثلاً تمام ارجمند اور بلنسنسل آپریٹرز بائسٹری آپریٹرز ہیں۔ منطقی آپریٹرز && اور    بھی بائسٹری آپریٹرز ہیں۔	یو زری آپریٹر صرف ایک آپریشن پر لاگو ہوتے ہیں جیسے منطقی NOT (!) آپریٹر کے پاس صرف ایک اور بینڈ ہوتا ہے۔ سائن آپریٹر (-) یو زری آپریٹر کی ایک اور مثال ہے۔ جیسے 5 - .

40: آپریٹرز کی ترجیح کیا ہے؟

جواب: اگر ایک پریشن میں ایک سے زیادہ آپریٹرز ہیں تو سوال یہ پیدا ہوتا ہے کہ پہلے کے حل کریں گے۔ اس مسئلے کو حل کرنے کے لیے ہر آپریٹر کو ترجیح دی گئی ہے۔ جس آپریٹر کی ترجیح زیادہ ہے وہ دوسرے آپریٹر سے پہلے حل ہو گا۔ اگر آپریٹر کی ایک جبی ترجیح ہو تو اسیں طرف والے آپریٹر سے پہلے باہم طرف والا آپریٹر حل کریں گے۔

41: ٹریزی آپریٹر کی وضاحت کریں؟

جواب: ہی پروگرامنگ لینگوچیج میں ایک ٹریزی آپریٹر (Ternary Operator) ہے جو تمیں اپریڈز پر کام کرتا ہے۔

42: scanf() اور getch() میں فرق کریں؟

scanf()	getch()
سی لینگوچیج میں ایک بلٹ ان فنکشن ہے جو صارف سے دیری ایسٹر میں ان پڑھ لیتا ہے۔ scanf فنکشن میں ہم فارمیٹ پیسناٹر کی مدد سے ان پڑھنیا کی موقع قسم کی وضاحت کرتے ہیں۔ اگر صارف ان شیجوں ناٹر میں دیتا ہے تو scanf میں بیان کردہ فارمیٹ پیسناٹر d\la\z \l\z ہونا چاہیے۔	فنکشن صارف سے ایک کریکٹر لینے کے لیے استعمال کیا جاتا ہے۔ صارف کا داخل کردہ کریکٹر سکرین پر ظاہر نہیں ہوتا ہے۔ یہ فنکشن عام طور پر پروگرام کی ایگزیکیوشن کو روکنے کے لیے استعمال کیا جاتا ہے کیونکہ پروگرام اس وقت تک آگئے نہیں چلتا جب تک کہ صارف کوئی ہنندہ بانے۔

43: مختلف آپریٹرز کی ترجیح بیان کریں؟

ترجیح	آپریٹر
1	( )
2	!
3	* , / , %
4	+ , -
5	> , < , >= , <=
6	== , !=
7	&&
8	
9	=

44: clrsqr () کا استعمال کھیسیں؟

جواب: یہ نکشن سی لینکوئچ ایڈٹر کی آؤٹ پٹ سکرین کو صاف کرنے کے لیے استعمال ہوتا ہے۔

Syntax: clrsqr ();

## یونٹ نمبر 3

## مشروط منطق

س: کنڑول شیئنٹش کیا ہیں؟ کنڑول شیئنٹش کی اقسام کے ہم لکھیں۔  
 کسی بھی مسئلے کو حل کرنے کے لیے کنڑول شیئنٹش کسی پروگرام چلنے کے بھاؤ کو کنڑول کرنے کے لیے استعمال کی جاتی ہیں۔ بعض اوقات  
 ہمیں کسی شرط کی بناء پر ہدایات (instructions) کے ایک سیٹ پر عمل درآمد کرنے کی ضرورت ہوتی ہے اور اگر یہ شرط درست نہ ہو تو ہدایات کے دوسرے  
 سیٹ پر عمل درآمد کرنا پڑتا ہے۔ مرید یہ کہ بعض اوقات ہمیں کسی بار شیئنٹش کے ایک حصے کو ایک خالص حد تک دھرا پڑتا ہے۔ C لینگوچ میں تین طرح کے  
 کنڑول شیئنٹش ہوتے ہیں۔

1. سیکوئیشل کنڑول شیئنٹش (Sequential Control Statements)
2. کنڈیشنل کنڑول شیئنٹش (Conditional Control Statements)
3. ریپیتیشن کنڑول شیئنٹش (Repetition Control Statements)

س: سلیکشن شیئنٹش کیا ہیں؟ میں سلیکشن شیئنٹش کی ضرورت کیوں ہے؟  
 وہ شیئنٹش جو ہمیں یہ فیصلہ کرنے میں مدد کرتی ہیں کہ کونسی شیئنٹش کو شرائط کی بنیاد پر پہلے چلایا جائے گا سلیکشن شیئنٹش کہلاتی  
 ہے۔ سلیکشن شیئنٹش کی دو اقسام ہوتی ہیں۔

If .1 شیئنٹ

if-else .2 شیئنٹ

س: اف شیئنٹ کیا ہے؟ اس کا ذہانچہ ہیان کریں۔  
 C لینگوچ میں If شیئنٹ کے ذریعے شرط بتا کر اس کو ایک کوڈ سے منسوب کیا جاتا ہے۔ اگر مخصوص شرط درست ثابت ہو تو کوڈ پر مسلسل درآمد ہو  
 جاتا ہے ورنہ کوڈ پر عمل درآمد نہیں ہوتا ہے۔ if شیئنٹ کے اندر والی شیئنٹ کو میں کے ذریعے یہ طرح سے انڈینٹ (Indent) کریں۔ اس سے پروگرام کو  
 سمجھنا آسان ہو جاتا ہے۔

if شیئنٹ کا ذہانچہ

C لینگوچ میں if شیئنٹ کا ذہانچہ یہ ہے۔

(شرط / کنڈیشن) if

منسوب کیا گیا کوڈ

یہاں if شیئنٹ کے عمومی ذہانچے میں شامل مختلف حصولوں کی مختصر تفصیل یہ ہے۔

1. دیئے گئے ذہانچے میں if ایک کی۔ ورثہ ہے جس کے بعد دو این ( ) کے اندر شرط لکھی جاتی ہے۔

## مشروع منطق

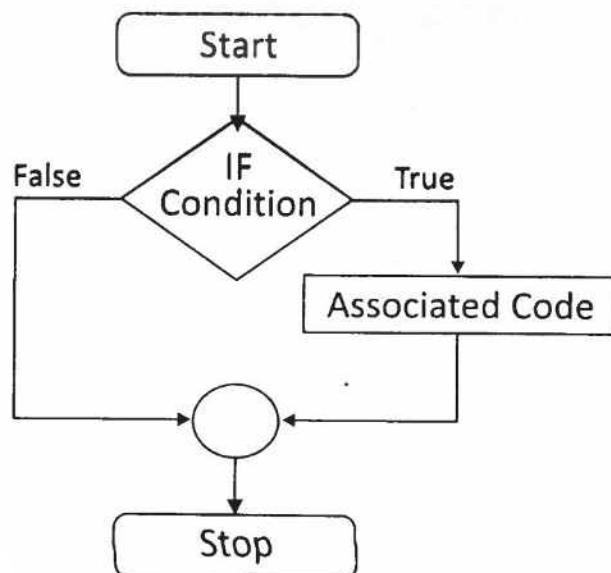
2. ایک شرط کوئی درست ایکپریشن ہو سکتی ہے جس میں ارتھمینٹک ایکپریشنز، ریلیشنل ایکپریشنز، منطقی ایکپریشنز یا ان کا مجموعہ شامل ہو سکتا ہے۔ درست ایکپریشنز کی چند مثالیں ہیں جنہیں بطور شرط استعمال کیا جاسکتا ہے۔

رزلٹ	ایکپریشن	نمبر شار
True		6 1
True	$6 + 3$	2
False	$6 - 6$	3
True	$6 > 5$	4
False	$6 == 4$	5
True	$!(5 > 6)$	6
False	$(6 > 5) \&& (10 < 9)$	7
True	$(6 > 5)    (10 > 9)$	8

کوئی بھی ایکپریشن جس کی قیمت صفر ہو وہ درست ہوتی ہے جیسے ایکپریشن  $1 = a$  اور  $1 = b = 0$  درست (True) ایکپریشن ہے لیکن ایکپریشن  $0 = c$  غلط (False) ہے۔ ایکپریشن میں دیری ایجل بھی استعمال کیے جاسکتے ہیں اس صورت میں دیری ایجل کی قیتوں کو استعمال کرتے ہوئے ایکپریشن کی صحیح / خاطط قیمت نکالی جاسکتی ہے۔

3. منسوب کردہ کوڈ لینگوچ کی کوئی بھی درست شیئٹشنس کا مجموعہ ہو سکتا ہے۔ اس میں ایک یا ایک سے زیادہ شیئٹشنس ہو سکتی ہیں۔

If شیئٹ کا فلوچارٹ:



اگر ہم ایک سے زیادہ شیئنٹس کو if شیئنٹ سے جوڑنا چاہتے ہیں تو ہمیں انھیں ایک {} بلاک (نیز می بریکنوں) کے اندر بند کرنے کی ضرورت ہے۔ لیکن اگر ہم صرف ایک شیئنٹ کو جوڑنا چاہتے ہیں تو یہ () بلاک (نیز می بریکنوں) کے اندر بند ہو سکتا ہے لیکن یہ لازمی نہیں ہے۔ یہ مندرجہ ذیل مثال سے ظاہر کیا جائے۔

```
#include<stdio.h>
void main()
{
    int a = 12;
    if (a % 2 == 0)
    {
        printf ("The variable a contains an even value.");
        printf("\nYou are doing a great job.");
    }
}
```

**Output**

The variable a contains an even value.  
You are doing a great job.

جب قیمت 12 کو 2 سے تقسیم کیا جاتا ہے تو 0 پچتا ہے۔ لہذا اگر if شیئنٹ کے اندر کی شرط درست ہے۔ اور چونکہ دونوں printf شیئنٹس {} بلاک (نیز می بریکٹ) کے اندر ہیں۔ لہذا دونوں شیئنٹس پر عملدرآمد ہو گا۔ مندرجہ ذیل پر غور کریں:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int a = 4;
    int b = 5;
    if( a > b)
        printf ("The value of a is greater than b.");
    printf("\nYou are doing a great job..");
    getch();
}
```

**Output**

You are doing a great job.

جیسا شرط درست نہیں ہے اور پہلی شیئنٹ جو 4 کا شیئنٹ {} بلاک (نیز می بریکنوں) کے اندر نہیں ہیں۔ لہذا صرف دوسرا شیئنٹ پر عمل ہو گا کیونکہ بغیر کسی {} بلاک (نیز می بریکنوں میں بند) کے صرف پہلی شیئنٹ کو 4 کا شیئنٹ کے ساتھ منسوب کیا جاتا ہے۔

## مشروط منطق

77

C لینگوئیج میں ایک پروگرام لکھیں جو طالب علم کی پرنسپل کو ان پٹ کے طور پر لیتا ہے اور اگر پرنسپل 50 سے اور ہے تو "PASS" دکھاتا ہے۔

### Program

```
#include<stdio.h>
#include<conio.h>
void main ()
{
    clrscr();
    float percentage;
    printf ("Enter the percentage:");
    scanf ("%f",&percentage);
    if (percentage > 50)
        printf ("PASS\n");
    getch();
}
```

### Output

Enter the percentage: 47

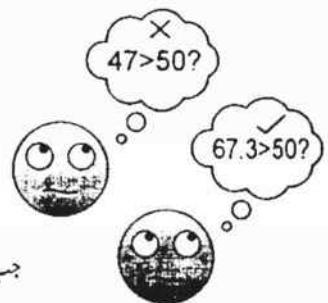
جب 47 ان پٹ دی گئی تو پروگرام ختم ہو گی کیونکہ  $47 < 50$  سے کم ہے اور شرط پوری نہیں ہوئی۔

Enter the percentage: 67.3

PASS

جب 67.3 ان پٹ دی گئی تو کنسول (console) پر "pass" پر نہ ہو اکیونکہ  $67.3 > 50$  سے بڑا ہے اور شرط

پوری ہو گئی۔



## مشروط منطق

78

س: ایک مکائن کرنے والی کمپنی اپنے ملازمین کی تنخواہ مندرجہ ذیل قارموں کی مدد سے بناتی ہے۔

$$\text{Gross Salary} = \text{Basic Salary} + (\text{Number of Items Sold} \times 8) + \text{Bonus}$$

اگر فروخت شدہ اشیاء کی تعداد 100 سے زیاد ہے اور ٹوٹنے والی اشیاء کی تعداد 0 ہے تو بونس (Bonus) 10000 روپے ہو گا۔ نہیں تو بونس 0 ہو گا۔ ایک پروگرام لکھیں جو Basic Salary, فروخت شدہ (No of Sold Items) اور ٹوٹنے والی اشیاء (No. of Broken Items) بطور صارف سے ان پڑ کے طور پر لے اور ملازم کی Gross Salary کا حساب پرداز کرے اور اس کا فلوقارٹ بھی بناتے۔

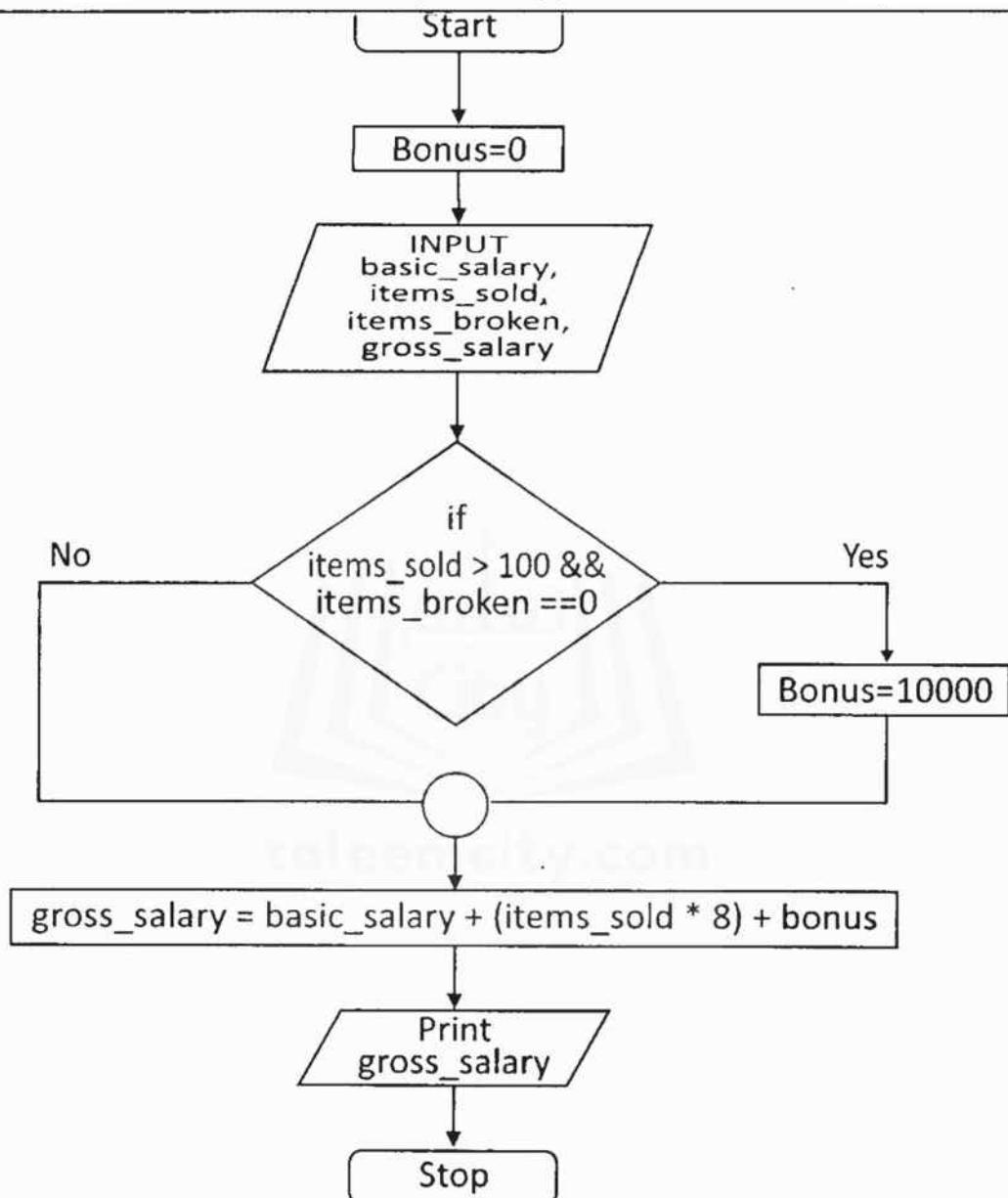
### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int basic_salary, items_sold, items_broken, gross_salary;
    int bonus = 0;
    printf("Enter the basic salary:");
    scanf("%d", &basic_salary);
    printf("Enter the number of items sold:");
    scanf("%d", &items_sold);
    printf ("Enter the number of items broken:");
    scanf("%d", &items_broken);
    if (items_sold > 100 && items_broken == 0)
        bonus = 10000;
    gross_salary = basic_salary + (items_sold * 8) + bonus;
    printf ("Gross salary of the employee is %d", gross_salary);
    getch();
}
```

### تفصیل

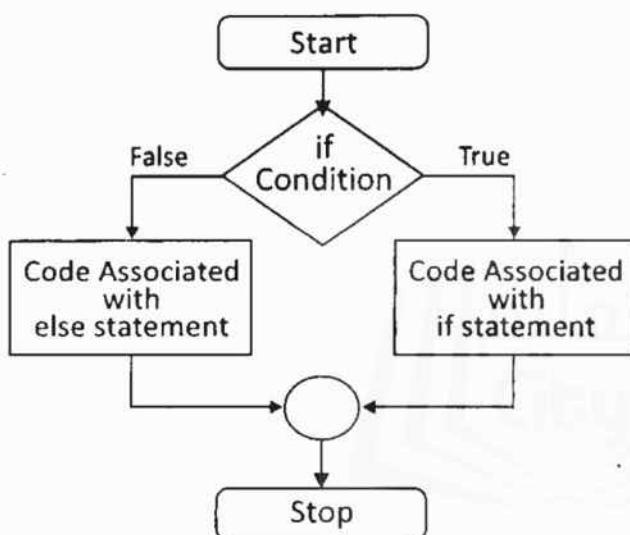
مندرجہ بالامثل میں بونس کو 0 سے انشلاہ کیا گیا ہے کیونکہ اگر فروخت شدہ اشیاء کی تعداد 100 سے کم ہو تو خود بخوبی بونس کو 0 سمجھا جائے۔ if سینٹٹ کے اندر یہ چیک کیا جاتا ہے کہ آیا فروخت شدہ اشیاء کی تعداد 100 سے زیاد ہے یا نہیں۔ اگر ایسا ہے تو بونس 10000 مقرر کر دیا جاتا ہے۔ واضح رہے کہ Gross Salary کا حساب کتاب if بلاک (نیز می بریکنوس) کے باہر شمار کیا گیا ہے کیونکہ فروخت شدہ اشیاء کی تعداد 100 سے زیاد ہے یا نہیں ہے تب بھی Gross salary کا حساب ہوتا ضروری ہے۔

## مشروع منطق



س: if-else شیئنٹ کیا ہے؟ اس کے ذہانی کی وضاحت کریں اور مثالیں دیں۔  
 if شیئنٹ میں اگر مطلوبہ شرط درست ہو تو if شیئنٹ سے منسوب کردہ شیئنٹس کے سیٹ پر عملدرآمد ہوتا ہے ورنہ else شیئنٹ سے منسوب کردہ شیئنٹس کے سیٹ پر عمل درآمد نہ ہوتا ہے۔  
 if-else شیئنٹ کا ذہانی چیز یہ ہے:

شیئنٹ کا فلوچارٹ if-else



(کٹھن) if

منسوب کردہ کوڈ

else

منسوب کردہ کوڈ

اگر کٹھن درست ہے تو if شیئنٹ سے منسوب کردہ کوڈ پر عمل درآمد ہو گا ورنہ else شیئنٹ سے منسوب کردہ کوڈ پر عمل درآمد ہو گا۔

مکن ہے کہ if شیئنٹ کے ساتھ else شیئنٹ نہ ہو لیکن else شیئنٹ کے لیے if شیئنٹ ہونا ضروری ہے۔  
 کی-ورڈ سے پہلے اگر if کے اندر ایک سے زیادہ شیئنٹس آرہی ہوں تو انہیں {} بلاک (بیز می بریکنوں میں بند) کے اندر بند ہونا چاہیے بصورت دیگر کپاٹر ایک ایر جاری کرتا ہے۔

#### Program:

```
#include<stdio.h>
void main ()
{
    int a = 15;
    if (a % 2 == 0)
        printf("The variable a contains an even value.");
        printf("\nYou are doing a great job.");
    else
        printf ("The variable a contains an odd value.");
}
```

Error

else without an associated if

## مشروط منطق

81

مذکورہ کوڈ کو کچھ نہیں کیا جاسکتا کیونکہ { } بلاک (نیز میں بریکٹوں) کے بغیر صرف ایک شرائطی if شرائط کے ساتھ ملک ہے۔ اس صورت میں پہلی شرائطی لینی ("if printf("The variable a contain an even value."") سے منسوب نہیں ہے لیکن دوسری شرائطی لینی ("else printf("\nyou are not doing a great job.")") سے منسوب نہیں ہے۔ اس لیے else والا حصہ بھی if شرائط سے ملک نہیں ہے۔ ہم جانتے ہیں کہ else کا if بلاک (نیز میں بریکٹوں) سے ملک ہوتا ضروری ہے۔ اس مسئلے کو حل کرنے کے لیے ہم کی وردہ else سے پہلے والی دونوں شرائط کو { } بلاک (نیز میں بریکٹوں) کے اندر رکھتے ہیں۔

### کپاڈٹ شرائطی بلاک (Compound Statement / Block)

بریکٹوں میں بند متعدد ہدایات کا ایک مجموعہ بلاک یا کپاڈٹ شرائط کہلاتا ہے۔ کپاڈٹ شرائط میں ایک عام ہی غلطی یہ ہے کہ ٹانپنگ کے دوران ایک یادو بریکٹوں کو بھول جانا ہے۔ اس ایرے سے بچنے کے لیے بہتر ہے کہ پہلے اپنگ اور کلوزگ بریکٹ ٹانپ کریں اور پھر بلاک میں شرائط ٹانپ کریں (نیز میں بریکٹوں میں بند)۔ کپاڈٹ شرائط کی مثال مندرجہ ذیل ہے۔

#### Program

```
#include<stdio.h>
#include<conio.h>
void main ()
{
    clrscr();
    int a = 15;
    if (a % 2 == 0)
    {
        printf("The variable a contains an even value.");
        printf("\nYou are doing a great job. ");
    }
    else
    {
        printf("The variable a contains an odd value.");
    }
    getch();
}
```

#### Output

The variable a contains an odd value.

ایک پروگرام کامیس جو ایک کریکٹر کو بطور ان پت لیتا ہے اگر صارف کا داخل کریکٹر 0 سے 9 کے درمیان ہندسہ ہو تو وہ "DIGIT" پر نت کرتا ہے ورنہ "NOT DIGIT" پر نت کرتا ہے۔

### Program

```
#include <stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    char input;
    printf ("Please enter a character :");
    scanf ("%c", &input);
    if (input >= '0' && input <= '9')
    {
        printf ("DIGIT\n");
    }
    else
    {
        printf ("NOT DIGIT\n");
    }
    getch();
}
```

### Output

Please enter a character : 5

DIGIT

اگر صارف ایک ہندسہ درج کرے 5:  
وہ سکرین پر DIGIT ظاہر ہو گا۔

Please enter a character : k

NOT DIGIT

اگر صارف کوئی اور کریکٹر درج کرتا ہے جیسے k پر  
"NOT DIGIT" ظاہر ہوتا ہے کیونکہ مطلوب شرط غلط ہو جاتی ہے۔

اگر if-سینٹنٹ یا else سینٹنٹ کے اندر ایک سے زیادہ بڑیات ہوں تو انھیں ایک بلاک (یعنی بریکنوں میں بند) کے اندر بند کریں۔ بصورت  
دیکھ کر مثلاً صرف ایک بڑیت کو اس سے منسوب کرتا ہے اور باقی بڑیات کو ان سے الگ سمجھتا ہے۔

کسی if-else-if کا لامانچ بیان کریں؟

C لینگوچ if-else-if سینٹنٹ بھی فراہم کرتی ہے جس کا لامانچ یہ ہوتا ہے۔

(کلی کنٹرین)

کوڈ پر مدل رآمد کیا جائے اگر کلی کنٹرین درست ہے۔

(دوسرا کنٹرین)

کوڈ پر مدل رآمد کیا جائے گا اگر کنٹرین 1 فلٹر ہے لیکن کنٹرین 2 درست ہے۔

N کنٹرین)

کوڈ پر مدل رآمد کیا جائے گا اگر تمام سابقہ کنٹرین ملا جیں لیکن کنٹرین N درست ہے۔

else

یہ کوڈ جب پڑے گا اگر تمام کنٹرین ملا جیں۔

س: ایک ایسا پروگرام لکھیں جو طالب علم کے نمبروں کی نیمہ کو ان پڑتے کے طور پر لے اور اس کا گزینہ دکھائے۔ درج ذیل بھیل گزینہ کی تفہیم کے معیار کو ظاہر کرتا ہے:

Percentage	Grade
80% and above	A
70% - 80%	B
60% - 70%	C
50% - 60%	D
Below 50%	F

**Program**

```
#include<stdio.h>
void main()
{
    clrscr();
    float percentage;
    printf ("Enter the percentage:");
    scanf ("%f, &percentage);
    {
        if (percentage >= 80)
            printf ("A\n");
        else if (percentage >= 70)
            printf ("B\n");
        else if (percentage >= 60)
            printf ("C\n");
        else if (percentage >= 50)
            printf ("D\n");
        else
            printf ("F\n");
    }
    getch();
}
```

## مشروط منطق

س: نیستہ سلیکشن سٹرکچر کو دھاخت سے بیان کریں؟

اگر جسم if-else کے ذہان پر کو غور سے دیکھتے ہیں جو کہ درج ذیل ہے۔

(کھٹکیں) if

منسوب کردہ کوڈ

else

منسوب کردہ کوڈ

یا else سینٹنٹ سے منسوب کردہ کوڈ شیئٹنٹس کا کوئی سیٹ ہو سکتا ہے جو C لینگوچ کے اعتبار سے درست ہو۔ اس کا مطلب یہ ہے کہ اگر if شیئٹنٹس یا if-else شیئٹنٹس میں ایک بلاک کے اندر جسم مزید شیئٹنٹس لکھ کرے ہیں۔ اس کا یہ بھی مطلب ہے کہ اندر والی if سینٹنٹ یا if-else شیئٹنٹ کے اندر ہم مزید اور شیئٹنٹس لکھ کرے اور یہ کام جتنی بار چاہے ہم دبرا سکتے ہیں۔ کندیشل شیئٹنٹس کے اندر کندیشل شیئٹنٹس نیستہ سلیکشن سٹرکچر کہلاتے ہیں۔ مندرجہ ذیل تمام ذہان پر نیستہ سلیکشن سٹرکچر سے منسوب کردہ کوڈ ہیں۔

Nested Selection Structure 1	Nested Selection Structure 2
<pre> if (condition1 is true)     if (condition 2 is true)         Associated code else     Associated code </pre>	<pre> if (condition1 is true)     if (condition2 is true)         Associated code else     if (conditions is true)         Associated code </pre>
Nested Selection Structure 3	Nested Selection Structure 4
<pre> if (condition1 is true)     if (condition2 is true)         Associated code else     Associated code else     if (conditions is true)         Associated code </pre>	<pre> if (condition1 is true)     if (condition2 is true)         Associated code else     Associated code else     if (condition3 is true)         Associated code else     Associated code </pre>

## مشروط منطق

85

س: مثال کی مدد سے نیشنل سلیکشن سرکپر کے استعمال کی وضاحت کریں؟

نیشنل سلیکشن سرکپر لینگوچ میں فیصلہ سازی کے لیے استعمال ہوتا ہے۔ نیشنل سلیکشن سرکپر کے استعمال کو سمجھنے کے لیے آئیے درج ذیل

مثال پر غور کریں۔

مثال: بجلی کے مل کی کمپنی اس فارمولے کے مطابق بجلی کے مل کا حساب لاتی ہے۔

$$\text{Bill Amount} = \text{Number of Units Consumed} \times \text{Unit Price}$$

بل کی رقم = استعمال شدہ یونٹ کی تعداد  $\times$  ایک یونٹ کی قیمت

بجلی کے صارفین کی دو حصیں ہیں یعنی کرشل اور گریڈ صارفین۔ گریڈ صارفین کے لیے یونٹ کی قیمت اس حساب سے رکھی جاتی ہے۔

Units Consumed	Unit Price
Units $<= 200$	Rs. 12
Units $> 200$ but Units $<= 400$	Rs. 15
Units $> 400$	Rs. 20

کرشل صارفین کے لیے ایک یونٹ کی قیمت اس حساب سے رکھی جاتی ہے۔

Units Consumed	Unit Price
Units $<= 200$	Rs 15
Units $> 200$ but Units $<= 400$	Rs 20
Units $> 400$	Rs 24

## مشروع منطق

86

ایک پروگرام لکھیں جو صارف کی قسم اور استعمال شدہ یونٹس کی تعداد ان پرٹے اور صارف کا ملک بتابے۔

### Program:

```
#include<stdio.h>
#include<conio.h>
void main ()
{
    clrscr();
    int units, unit_price, bill;
    char user_type;
    printf("Please enter h for- home user and c for commercial user: ");
    scanf("%c", &user_type);
    printf ("Please enter the number of units consumed: ");
    scanf("%d", &units);
    if (units <= 200)
        if (user_type == 'h')
            unit_price = 12;
        else if (user_type == 'c')
            unit_price = 15;
    else if (units > 200 && units <= 400)
        if (user_type == 'h')
            unit_price = 15;
        else if (user_type == 'c')
            unit_price = 20;
    else
        if (user_type == 'h')
            unit_price = 15;
        else if (user_type == 'c')
            unit_price = 24;
    bill = unit * unit_price;
    printf("Your electricity bill is %d", bill);
    getch();
}
```

س: ایک پروگرام لکھیں جو تین نمبروں میں سے بڑا نمبر دکھائے۔

### Program

```
include <stdio.h>
void main()
{
    int n1, n2, n3;
    printf ("Enter three numbers");
    scanf ("%d%d%d", &n1, &n2, &n3);
    if (n1 > n2 && n1 > n3)
        printf ("The largest number is %d", n1);
    else if (n2 > n3 && n2 > n1)
        printf ("The largest number is %d", n2);
    else
        printf ("The largest number is %d", n3);
}
```

س: ایک ایسا پروگرام لکھیں جو صارف کے اختیاب کے مطابق کوہ سانڈر اور دائیے کے جمکا حساب کئے۔

```
#include<stdio.h>
#include<conio.h>
void main ()
{
    clrscr();
    int choice;
    float volume;
    printf ("Find Volume\n");
    printf ("1. Cube\n2.Cylinder\n3.Sphere\nEnter your choice:");
    scanf ("%d", &choice);
    if (choice == 1)
    {
        float length;
        printf ("Enter Length: ");
        scanf ("%f ",&length);
        volume = length * length * length;
        printf ("Volume is %f ", volume);
    }
    else if (choice == 2)
    {
        float length1, radius1;
        printf ("Enter Length: ");
        scanf ("%f ", &length1);
        printf ("Enter Radius: ");
        scanf ("%f ", &radius1);
        volume = 3.142 * radius1 * radius1 *length1;
        printf ("Volume is %f ", volume);
    }
    else if (choice == 3)
    {
        float radius;
        printf ("Enter Radius: ");
        scanf ("%f ", &radius);
        volume = 3.142 * radius * radius * radius;
        printf ("Volume is %f ", volume);
    }
    else
    {
        printf ("Invalid Choice");
    }
    getch();
}
```

## مشق

س 1: کمیر الامتحانی سوالات

نمبر	سوال	D	C	B	A
1	کندیشل لا جک _____ میں مددویتی ہے۔	تمام	مزیور سنگ	مکاروں	فیصلوں
2	سینٹنس باتی ہیں کہ پروگرام کی سینٹنس کس ترتیب سے اجرا کیوں ہوں گی۔	تمام	کنڑول	مشروط	لوپ
3	If سینٹنس میں اگر کندیشن غلط ہے تو کیا ہوتا ہے؟	کپاٹر کندیشن بدلے کا مطالبہ کرتا ہے	بانی کوڈ چلنے لگتا ہے	انڈیکس آؤٹ آف باؤنڈ ایر آتا ہے	پروگرام رک جاتا ہے
4	int a = 5; if (a < 10) a++; else if (a > 4) a--;  ان میں سے کوئی سینٹنس چلے گی؟	کوئی نہیں	پہلی دونوں	a--;	a ++;
5	مندرجہ ذیل میں سے کون سی کندیشن یہ باتی ہے کہ a کا فیکر ہے یا نہیں؟	a + c == 0	a * c == 0;	c % a == 0	a % c == 0.
6	ایک کندیشن کوئی بھی ایک پریشن ہو سکتی ہے۔	تمام	لا جیکل	ریلیشن	ار تمیک
7	اگر if سینٹنس کے اندر ایک اور if سینٹنس ہو تو یہ سڑ کچھ کھلاتا ہے۔	ذکر پوزڈ	ریسینڈ	بوکسڈ	نیسٹڈ
8	تو یہ میں بند ایک سے زیادہ ایات کا سیٹ کھاتا ہے۔	حوب	بلاک	لٹ	بوکس

جواب

1	2	3	4	5	6	7	8
A	C	C	A	A	D	A	C

س2: درج ذیل کی وضاحت کریں۔

- صلو نمبر 74 پر ملاحظہ فرمائیں  
صلو نمبر 89 پر ملاحظہ فرمائیں  
صلو نمبر 89 پر ملاحظہ فرمائیں  
صلو نمبر 110 پر ملاحظہ فرمائیں  
صلو نمبر 84 پر ملاحظہ فرمائیں

- (1) کنڑول شیئٹش
- (2) کندیشل شیئٹش
- (3) سیکوئیشل کنڑول
- (4) کندیش
- (5) نیمنڈ سلیکشن سڑک پر

س3: درج ذیل سوالات کا مختصر جواب دیں۔

- 1) میں کٹھ پیش شیٹش کی ضرورت کہوں ہوتی ہے؟
  - 2) یکو پیش اور کٹھ پیش شیٹش میں فرق کر پ۔

سکونیشل شیمنش	کنٹنیشنل شیمنش
<p>سکونیشل کنٹرول C لینکو گھمیں ڈیفالٹ کنٹرول سترکچر ہے۔ سکونیشل کنٹرول کے مطابق تمام سیسیمینشنس ٹوڈی گئی ترتیب میں ایکزیکوٹ کیا جاتا ہے۔</p>	<p>”سیمینشس جو ہمیں یہ فیلڈ کرنے میں مدد ہوتی ہیں کہ کون سے سیمینشس کو کنٹرول کی بنیاد پر آگے چلا جائے گا کنٹرول سیمینش کہلاتی ہیں۔</p>

صفہ نمبر 74.80 پر ملاحظہ فرمائیں

صلوٰ نمبر 84 پر ملاحظہ فرمائیں

صفحہ نمبر 74 بر طبق فرمائیں

س ۴: کوڑے کے ان حصوں میں ایر رز کی جلاش کرس۔ فرض کریں کہ تمام ورثی ایمیل بیلے سے ہی ڈیکیٹر کیے ہوئے ہیں۔

a) if ( $x = 10$ )  
    printf ("Good");

**Error: Invalid sign = in C language.**

b) if ( $a < b \&\& b < c$ );  
          sum =  $a + b + c$ ;  
      else  
          multiply =  $a * b * c$ ;

**Error:** ; is not used in if statement  
(Syntax Error).

c) if ( $a < 7 < b$ )  
    printf ("7");

Error: 7 is printing as string.

d) if ( $a == b \& x == y$ )  
    flag = true;  
else  
    flag = false;

```
e) if (sum == 60 || product == 175)
    printf ("Accepted %c", sum);
else
    if (sum >=45 || product > 100)
        printf ("Considered %d", sum);
else
    printf ("Rejected")
```

**Error:** Double quotes are missing at  
printf ("Accepted %c", sum); and %d  
should be used instead of %c.

رس 5: مندرجہ ذیل کوڈ کی آٹھ پٹھ تائیں۔

```
a) int a = 7, b = 10;
a = a + b;
if ( a> 20 && b < 20 )
    b = a + b;
printf ("a = %d, b = %d", a, b);
```

**Output:** a=17, b=10

```
b) int x = 45;
if (x + 20 * 7 == 455)
    printf ("Look's Good");
else
    printf ("Hope for the Best");
```

**Output:** Hope for the Best

```
c) char c1 = 'Y', c2 = 'N';
int n1 = 5, n2 = 9;
n1 = n1 + 1;
c1 = c2;
if (n1 == n2 && c1 == c2)
    printf ("%d = %d and %c = %c", n1, n2, c1, c2);
else
    if (n1 < n2 && c1 == c2)
        printf ("%d < %d and %c = %c", n1, n2, c1, c2);
else
    printf ("Better Luck Next Time!");
```

**Output:** 6<9 and N=N

d) int a = 34, b = 32, c = 7, d = 15;  
 a = b + c + d;  
 if (a < 100)  
 a = a \* 2;  
 b = b \* c;  
 c = c + d;  
 if (a > b && c == d)  
 {  
 c = d;  
 b = c;  
 a = b;  
 }  
 else if (a > b && c > d || b >= d + c)  
 {  
 d = c \* c;  
 a = b \* b;  
 }  
 printf ("a=%d, b=%d, c=%d, d=%d", a, b, c, d);

Output: a=15360, b=224, c=22, d=484

e) int x = 5, y = 7, z = 9;  
 if (x % 2 == 0)  
 x++;  
 else  
 x = y + z;  
 printf (" x = %d\n", x);  
 if (x % 2 == 1 && y % 2 == 1 && z % 2 == 1)  
 printf ("All are Odd");  
 if (x > y || x < z)  
 {  
 if (x > y)  
 y++;  
 else  
 if (x < z)  
 x++;  
 }  
 printf ("x = %d, y = %d, z = %d", x, y, z);

Output: x=6, y=7, z=9

کس ۱: ایک پروگرام تسلی بودوا میر (int) و بہور ان پت یا ہمیسے اگر بنا گائے کہ میں پڑھا برد مرے برے سرتھی ہے۔

## **Program**

```
#include<stdio.h>
#include<conio.h>
#include <math.h>
int main()
{
    clrscr();
    int n1, n2;
    printf( "Input the first integer : " );
    scanf("%d", &n1);
    printf( "Input the second integer: " );
    scanf("%d", &n2);
    if(n1% n2 == 0)
        printf( "\n %d is a Factor of %d. \n", n1, n2 );
    else
        printf( "\n %d is not a factor of %d. \n", n1, n2 );
    getch();
}
```

## Output

```
Input the first integer : 9  
Input the second integer : 3  
9 is a Factor of 3.
```

مثلاً: ایک پروگرام لکھیں جو ایک نمبر کو بطور انپٹ لے اور اگر ان پٹ نمبر 3 کا حاصل ضرب ہو اور اس کا اکائی مقام (unit price) 5 پر 7 ہو تو "Yes" پر نت کرے۔ e.g: 15,75.

## Program

```
#include<stdio.h>
#include<conio.h>
#include <math.h>
int main()
{
    clrscr();
    int n1;
    printf( "Input Any number :" );
    scanf("%d", &n1);
    if(n1%3==0)
        printf( "YES %d ,%d",n1,n1*5);
    getch();
}
```

## Output

Input Any number : 15  
YES 15 , 75

شیخ 3: "گروری مارت" (Grocery Mart) میں دستیاب ڈکاؤنٹ کی فہرست درج ذیل ہے۔

Total Bill	Discount
1000	10%
2500	20%
5000	35%
10000	50%

ایک پروگرام لکھیں جو ان پٹ کے طور پر کل میں لیتا ہے اور بتاتا ہے کہ صاف کوتاڈ ڈکاؤنٹ ملائے اور ڈکاؤنٹ کے بعد قیمت کیا ہے۔

#### Program

```
#include<stdio.h>
#include<conio.h>
#include <math.h>

int main()
{
    clrscr();
    int total_bill;
    printf( "Enter your Total Bill : " );
    scanf("%d", &total_bill);
    if(total_bill >=1000 && total_bill <= 2499)
        printf("Discount = %d\nDiscounted Price =%f ",10,total_bill*0.1);
    else if(total_bill >= 2500 && total_bill <= 4999)
        printf("Discount = %d\nDiscounted Price =%f ",20,total_bill*0.2);
    else if(total_bill >= 5000 && total_bill <= 9999)
        printf("Discount = %d\nDiscounted Price =%f ",35,total_bill*0.35);
    else if(total_bill >= 10000)
        printf("Discount = %d\nDiscounted Price =%f ",50,total_bill*0.5);
    getch();
}
```

#### Output

```
Enter your Total Bill : 10000
Discount = 50
Discount Price = 5000.000000
```

## مشروط منطق

94

مشق 4: ایک پروگرام لکھیں جو پرداز کت کی اصل قیمت اور قیمت فروخت بلور ان پٹ کے طور پر لیتا ہے اور بتاتا ہے کہ پرداز کت لفج یا نصان پر فروخت ہوئی ہے۔ اور پروگرام کا لفج / نصان بھی بتائے۔

### Program

```
#include <stdio.h>
#include<conio.h>
#include <math.h>
int main()
{
    clrscr();
    int cp,sp, amt;
    /* Input cost price and selling price of a product */
    printf("Enter cost price: ");
    scanf("%d", &cp);
    printf("Enter selling price: ");
    scanf("%d", &sp);
    if(sp>cp)
    {
        /* Calculate Profit */
        amt = sp - cp;
        printf("Profit = %d", amt);
    }
    else if(cp>sp)
    {
        /* Calculate Loss */
        amt = cp - sp;
        printf("Loss = %d", amt);
    }
    else
    {
        /* Neither profit nor loss */
        printf("No Profit No Loss.");
    }
    getch();
}
```

### Output

Enter cost price : 1000

Enter selling price : 1500

Profit = 500

$$\text{hypotenuse}^2 = \text{base}^2 + \text{height}^2.$$

$$\text{ہے س مل } ^2 (\omega) = (بیار)^2 + (اویحائی)^2$$

## Program

```
#include <stdio.h>
#include<conio.h>
#include <math.h>
int main()
{
    clrscr();
    int x,y,z;
    printf("Input the three sides of a triangle :\n");
    scanf("%d %d %d",&x,&y,&z);
    if((x*x) + (y*y) == (z*z) || (x*x) + (z*z) == (y*y) || (y*y) + (z*z) == (x*x) )
    {
        printf("It is a right angle triangle!\n");
    }
    else
    {
        printf("It is not a right angle triangle!\n");
    }
    getch();
}
```

## Output

**Input the three sides of a triangle :**

7

3

5

It is not a right angle triangle!

میں 6: ایک آئی ٹائج نوورٹی میں داخلے کے لئے الیٹ کا معیار یہ ہے۔

- میزک میں کم از کم 60 نمبر۔

- انٹر میڈیٹ (پری انچیئرنگ یا آئی سی ایس) میں کم از کم 65 نمبر

- داخلہ نیٹ میں کم از کم 65 نمبر

ایک پروگرام لکھیں جو میزک، انٹر میڈیٹ اور داخلہ نیٹ کے حاصل کردہ اور کل نمبر انپٹ کے طور پر لے اور پروگرام کو ہتھے کر طالب علم اعلیٰ ہے ।  
تمہارے

### Program

```
#include <stdio.h>
#include<conio.h>
#include <math.h>
int main()
{
    clrscr();
    int omm,tmm,omi,tmi,ome,tme;
    printf("Enter Your obtained marks in Matric :");
    scanf("%d",&omm);
    printf("Enter Total marks in Matric");
    scanf("%d",&tmm);
    printf("Enter Your obtained marks in Intermediate :");
    scanf("%d",&omi);
    printf("Enter Total marks in Intermediate :");
    scanf("%d",&tmi);
    printf("Enter Your obtained marks in Entrance :");
    scanf("%d",&ome);
    printf("Enter Total marks in Entrance :");
    scanf("%d",&tme);
    if(((omm*100) / tmm) > 60 && ((omi*100) / tmi ) > 65 && ((ome*100) / tme ) > 65)
    {
        printf("You are Eligible for Admission");
    }
    else
    {
        printf("Not Eligible");
    }
    getch();
}
```

### Output

Enter Your obtained marks in Matric : 833
Enter Total Marks in Matric : 1050
Enter Your obtained marks in Intermediate : 804
Enter Total Marks in Intermediate : 1100
Enter Your obtained marks in Entrance : 72
Enter Total Marks in Entrance : 100
Your are Eligible for Admission

## مشروط منطبق

شیء 7: ایک پرограм لصیس جو دینے کے نیل کی باتا پر باتاے کہ ملازم کو کتنا بونس ملے گا۔

Salary	Experience with Company	Bonus Tasks	Bonus
10000	2 Years	5	1500
10000	3 Years	10	2500
25000	3 Years	4	2000
75000	4 Years	7	3500
100000	5 Years	10	5000

پرограм ملازم کی تجواہ، جگہ اور بونس ناٹکس کی تعداد ان پہلے اور سکرین پر بونس دکھائے۔

### Program

```
#include <stdio.h>
#include<conio.h>
#include <math.h>
int main()
{
    clrscr();
    int salary,experience,bonus_task;
    printf("Enter Your salary: ");
    scanf("%d",&salary);
    printf("Enter Your Experience: ");
    scanf("%d",&experience);
    printf("Enter Your Bonus task: ");
    scanf("%d",&bonus_task);
    if(salary == 10000 && experience == 2 && bonus_task == 5)
        printf("Bonus is = %d",1500);
    if(salary == 10000 && experience == 3 && bonus_task == 10)
        printf("Bonus is = %d",2500);
    if(salary == 25000 && experience == 3 && bonus_task == 4)
        printf("Bonus is = %d",2000);
    if(salary == 75000 && experience == 4 && bonus_task == 7)
        printf("Bonus is = %d",3500);
    if(salary == 100000 && experience == 5 && bonus_task == 10)
        printf("Bonus is = %d",5000);
    getch();
}
```

### Output

```
Enter your salary: 10000
Enter Your Experience: 2
Enter your Bonus task: 5
Bonus is = 1500
```

## سرگرمیوں کا حل

سرگرمی 3.1: ایسا پروگرام لکھیں جو کسی شخص کی عمر (Age) ان پتھ کے طور پر لے اور سکرین پر "Teenager" ظاہر کرے اگر عمر 13 سے 19 سال کے درمیان ہو۔

### Solution:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int teenager;
    printf("Enter your age: ");
    scanf("%d", &teenager);
    if(teenager >= 13 && teenager <= 19)
    {
        printf("Teenager");
    }
    else
        printf("Not Teenager");
    getch();
}
```

### Output:

Enter your age: 16

Teenager

## مشروع منطق

99

مرگری 3.2: ایک پروگرام لکھیں جو سال کو ان پٹ کے طور پر لے۔ اگر یہ سال یا (Leap) ہو تو سکرین پر "Leap Year" ظاہر کرے۔ لیکن کام وہ سال ہوتا ہے جو 42 پر مکمل تقسیم ہو جائے۔

**Solution:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int year;
    printf("Enter Year : ");
    scanf("%d", &year);
    if(year%4==0)
    {
        printf("Leap Year");
    }
    else
        printf("Not Leap Year");
    getch();
}
```

**Output:**

Enter Year : 1988

Leap Year

مرگی 3.3: ایک پروگرام لکسیں جو ایک شخص کے جسم کا درجہ حرارت ان پڑے اور اگر ان پٹ 98.6 سے زیاد ہو تو "You have fever" پر نت کرے ورنہ "You don't have fever" پر نت کرے۔

**Solution:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    float temp;
    printf("Enter your body temperature : ");
    scanf("%f ", &temp);
    if(temp > 98.6)
    {
        printf("You have fever");
    }
    else
        printf("You don't have fever");
    getch();
}
```

**Output:**

Enter your body temperature : 98.7

You have fever

سرگزی 3.4: ایک یونیورسٹی کے اندر گرینجھٹ پروگرام کے لئے الہیت کا معیار یہ ہے۔  
 BSSE پروگرام: اندر میڈیٹ میں 80% یا اس سے زیادہ نمبر  
 BSCS پروگرام: اندر میڈیٹ میں 70% یا اس سے زیادہ نمبر  
 BSIT پروگرام: اندر میڈیٹ میں 60% یا اس سے زیادہ نمبر  
 درست طالب علم یونیورسٹی کے کسی بھی پروگرام میں داخلہ نہیں لے سکتا۔  
 ایک پروگرام کیسیں جو اندر میڈیٹ کے نمبروں کی پرستی کے ارتباً کے طالب علم کن کون پروگرام میں درخواست دینے کی الہیت رکھتا ہے۔

**Solution:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int per;
    printf("Enter your intermediate percentage : ");
    scanf("%d", &per);
    if(per >= 80)
        printf("Your are eligible to get admission in BSSE, BSCS and BSIT");
    else if(per >= 70)
        printf("Your are eligible to get admission in BSCS and BSIT");
    else if(per >= 60)
        printf("Your are eligible to get admission in BSIT");
    else
        printf("Your are not eligible to get admission in University");
    getch();
}
```

**Output:**

Enter your intermediate percentage : 85

Your are eligible to get admission in BSSE, BSCS and BSIT.

سرگزی 3.5: ایک پر گرام کسیں دو اشجار ان پٹ لے اور صارف سے کہے کہ 1 سے 4 کے درمیان اپنا انتساب درج کرے۔ پھر پر گرام دیئے جو محل کے مطابق اور یعنی انجام دے اور نتیجہ بتائے۔

Choice	Operation
1	Addition
2	Subtraction
3	Multiplication
4	Division

**Solution:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int a,b,c;
    printf("Enter 1st integer : ");
    scanf("%d", &a);
    printf("Enter 2nd integer : ");
    scanf("%d", &b);
    printf("Enter 1 for Addition:\n Enter 2 for Subtraction:\n");
    printf("Enter 3 for Multiplication:\n Enter 4 for Division: \nEnter choice");
    scanf("%d", &c);
    if(c == 1)
        printf("%d + %d = %d", a , b , a + b);
    else if(c == 2)
        printf("%d - %d = %d", a , b , a - b);
    else if(c ==3)
        printf("%d * %d = %d", a , b , a * b);
    else if(c ==4)
        printf("%d / %d = %d", a , b , a / b);
    else
        printf("Please enter value between 1 to 4");
    getch();
}
```

Enter 1st integer : 9

Enter 2nd integer : 3

Enter 1 for Addition:

Enter 2 for Subtraction:

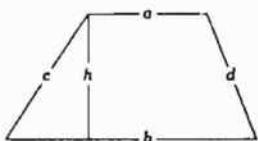
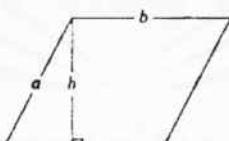
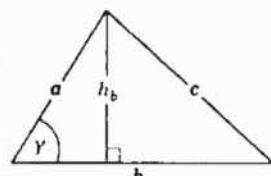
Enter 3 for Multiplication:

Enter 4 for Division:

Enter choice 1

 $9 + 3 = 12$ 

مرجی 3.6: ایک پروگرام لکسیں جو صارف کے انتخاب کے مطابق مثلث، متوازی الاضلاع، مکعب یا تریم کا رقبہ معلوم کرے اور سکرین پر دکھائے۔

(مربع) Area of trapezium	Area of parallelogram (متوازی الاضلاع)	(مثلث) Area of triangle														
$(a + b) \times h \times \frac{1}{2}$	$\text{base} \times \text{height}$	$\text{base} \times \text{height} \times \frac{1}{2}$														
 $A = 8$ <table border="1"> <tr> <td>a Base</td> <td>2</td> </tr> <tr> <td>b Base</td> <td>2</td> </tr> <tr> <td>h Height</td> <td>4</td> </tr> </table> Solution $A = \frac{a+b}{2} h = \frac{2+2}{2} \cdot 4 = 8$	a Base	2	b Base	2	h Height	4	 $A = 6$ <table border="1"> <tr> <td>b Base</td> <td>2</td> </tr> <tr> <td>h Height</td> <td>3</td> </tr> </table> Solution $A = b h = 2 \cdot 3 = 6$	b Base	2	h Height	3	 $A = 4$ <table border="1"> <tr> <td>b Base</td> <td>2</td> </tr> <tr> <td>h_b Height</td> <td>4</td> </tr> </table> Solution $A = \frac{h_b b}{2} = \frac{4 \cdot 2}{2} = 4$	b Base	2	h_b Height	4
a Base	2															
b Base	2															
h Height	4															
b Base	2															
h Height	3															
b Base	2															
h_b Height	4															

**Solution:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int choice;
    float area;
    printf("Find Area\n");
    printf("1.Triangle\n2.Parallelogram\n3.Trapezium\nEnter your choice:");
    scanf("%d", &choice);
    if(choice == 1)
    {
        float b , h;
        printf("Enter base of triangle:");
        scanf("%f ", &b);
        printf("Enter height of triangle:");
        scanf("%f ", &h);
        area = b * h * 1 / 2;
        printf("Area of Triangle:%f ",area);
    }
    if(choice == 2)
    {
        float b , h;
        printf("Enter base of parallelogram:");
        scanf("%f ", &b);
        printf("Enter height of parallelogram:");
        scanf("%f ", &h);
```

```

        area = b * h ;
        printf("Area of Parallelogram:%f ",area);
    }
    if(choice == 3)
    {
        float b1 , b2, h;
        printf("Enter base1 of trapezium:");
        scanf("%f ", &b1);
        printf("Enter base2 of trapezium:");
        scanf("%f ", &b2);
        printf("Enter height of trapezium:");
        scanf("%f ", &h);
        area = (b1 + b2) * h * 1 / 2;
        printf("Area of Trapezium:%f ",area);
    }
    else
        printf ("Wrong Entry. Please enter value between 1 to 3.");
    getch();
}

```

**Output:**

Find Area

1.Triangle

2.Parallelogram

3.Trapezium

Enter your choice: 1

Enter base of triangle: 2

Enter height of triangle: 3

Area of Triangle: 3.000000

## کشہر الاتختابی سوالات

نمبر	سوال	A	B	C	D
1	ہم پر گرام پٹنے کے بھاؤ کو _____ کے ذریعے کنڑول کر سکتے ہیں۔	کنڑول سینٹنس	سیکونٹیشن کنڑول	سینٹنس	ریٹیٹیشن کنڑول سینٹنس
2	C لینگوچ میں _____ قسم کی کنڑول سینٹنس ہیں۔	2	3	5	7
3	درج ذیل میں سے کون سا کنڑول سینٹس کی قسم نہیں ہے؟	سینٹنس	سینٹنس	ریٹیٹیشن کنڑول سینٹنس	پیک کنڑول سینٹنس
4	وہ سینٹنس جو ہمیں یہ فیصلہ کرنے میں مدد کرتی ہیں کہ کون سے سینٹنس کو شراکٹ کی بنیاد پر آگے چلایا جائے گا کہلاتی ہیں۔	سینٹنس	سینٹنس	سینٹنس	پیک کنڑول سینٹنس
5	سینٹس کی اقسام ہیں۔	if	If - else سینٹس	دونوں A اور B	سینٹس
6	C لینگوچ میں ----- کے ذریعے شرط بنا کر اس کو ایک کوڈ سے منسوب کیا جاتا ہے۔	سینٹس	if	If - else سینٹس	سینٹنس
7	اگر مخصوص شرط ----- ثابت ہو تو کوڈ پر عملدرآمد ہو جاتا ہے ورنہ کوڈ پر عمل درآمد نہیں ہوتا ہے۔	غلط	درست	دونوں A اور B	کوئی بھی نہیں
8	----- سینٹس کا ڈھانچہ ہے؟	(کندیشن) منسوب کیا گیا کوڈ	کندیشن	Else کندیشن	If - else کندیشن منسوب کیا گیا کوڈ
9	if سینٹس کے ڈھانچے میں ایک کی وردہ بے جس کے بعد _____ کے اندر ایک کندیشن آتی ہے۔	(.)	<>	{}	[]
10	_____ کے مطابق تمام سینٹس کو دیے گئے سینٹس میں ایکسیکوٹ کیا جاتا ہے۔	سینٹنس کنڑول	ریٹیٹیشن کنڑول	سینٹنس	ریٹیٹیشن کنڑول

یہ سب	ریپیشل کنڑول	سیکونٹیشل کنڑول	چیک کنڑول	سی لینگوئیں ڈیفائل کنڑول کا ڈھانچہ ہے۔	11
If - else شیمنٹ	If شیمنٹ	کنڈیشنل شیمنٹ	کنڑول شیمنٹ	اگر شیمنٹ درست ہے تو اس شیمنٹ کے تحت شیمنٹس کے مجموعہ کو اگر یہ شرط کیا جاتا ہے	12
کپاؤند شیمنٹ، بلک	کوڈ، بلک	بلک، ریپیشل شیمنٹ	بلک، کپاؤند شیمنٹ	ایک سے زیادہ ہدایات کا مجموعہ جو بریکوں میں بند ہے اسے یا کہا جاتا ہے۔	13
کوڈ	بلک	قوسین	بوکس	اگر ایک سے زیادہ ہدایات if شیمنٹ یا if - else شیمنٹ کے تحت ہیں تو انہیں کی محل میں بند کریں۔	14
If نیمنڈ سلیکشن	If - else شرکر	نیمنڈ سلیکشن شرکر	کنڈیشنل شرکر	کنڈیشنل شیمنٹس کے اندر کنڈیشنل شیمنٹس کو کہا جاتا ہے۔	15
ریپیشل شیمنٹ	سلیکشن شیمنٹ	کنڑول شیمنٹ	کپاؤند شیمنٹ	میں ٹائپنگ کے دوران ایک یا دو بریکوں کو بھول جانا ایک عام ہی غلطی ہے۔	16

جوابات

1	2	3	4	5	6	7	8	9	10
A	B	D	C	C	B	B	A	A	C
11	12	13	14	15	16				
B	D	A	C	B	A				

### مختصر سوالات

1: آپ کنڑول شیمنٹ کے بارے میں کیا جانتے ہیں؟

جواب: کسی بھی مسئلے کو حل کرنے کے لیے کنڑول شیمنٹس کسی پروگرام چلنے کے بعد کو کنڑول کرنے کے لیے استعمال کی جاتی ہیں۔ بعض اوقات ہمیں کسی شرط کی بناء پر ہدایات (instructions) کے ایک سیٹ پر عمل درآمد کرنے کی ضرورت ہوتی ہے اور اگر یہ شرط درست نہ ہو تو ہدایات کے دوسرا سے سیٹ پر عمل درآمد کرنا پڑتا ہے۔

2: کنڑول شیمنٹس کی کتنی اقسام ہیں؟

جواب: C لینگوئیں میں تین طرح کے کنڑول شیمنٹس ہوتے ہیں۔

3. ریپیشل کنڑول شیمنٹس

(Repetition Control Statements)

2. کنڈیشنل کنڑول شیمنٹس

(Conditional Control Statements)

1. سیکونٹیشل کنڑول شیمنٹس

(Sequential Control Statements)

## مشروط منطق

108

3: سیکونسٹیشن کنڑول کی تعریف کریں؟

جواب: سیکونسٹیشن کنڑول سی لینگوچ میں ڈیفالٹ کنڑول سر کچھ ہے۔ سیکونسٹیشن کنڑول کے مطابق تمام شیئمنش کو دی گئی ترتیب پر اگر بکیوٹ کیا جاتا ہے۔  
4: سلیکشن شیئمنش کیا ہے؟

جواب: وہ شیئمنش جو ہمیں یہ فیصلہ کرنے میں مدد کرتی ہیں کہ کوئی شیئمنش کو شرائط کی بنیاد پر پہلے چلا جائے گا سلیکشن شیئمنش کہلاتی ہیں۔

5: سلیکشن شیئنٹ کی کتنی اقسام ہیں؟

جواب: If-else (1) شیئنٹ If-then (2)

جواب: سلیکشن شیئنٹ کی دو اقسام ہیں۔

6: if شیئنٹ کا ذہانچہ بیان کریں؟

جواب: C لینگوچ میں if شیئنٹ کے ذریعے شرط بتا کر اس کو ایک کوڈ سے منسوب کیا جاتا ہے۔ اگر مخصوص شرط درست ثابت ہو تو کوڈ پر عملدرآمد ہو جاتا ہے درن کوڈ پر عملدرآمد نہیں ہوتا ہے۔

7: if شیئنٹ کا ذہانچہ بیان کریں؟

جواب: C لینگوچ میں if شیئنٹ کا ذہانچہ یہ ہے۔

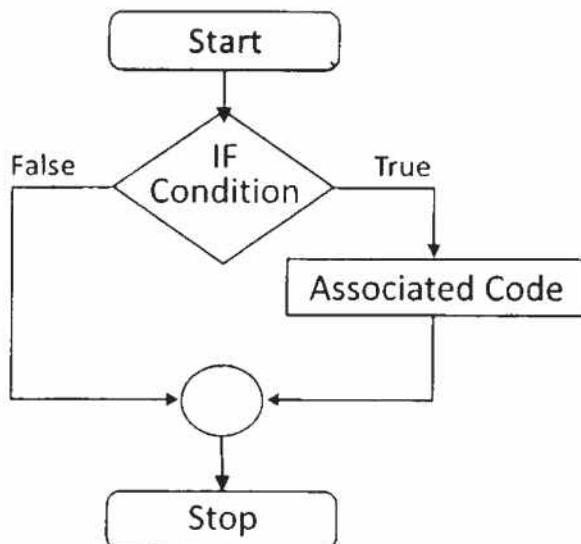
(شرط/اکنڈیشن) if

منسوب کیا گیا کوڈ

8: if شیئنٹ کے ذہانچہ کا مقدمہ بیان کریں؟

جواب: if ایک کی۔ ورثہ ہے جس کے بعد دو اور ( ) کے اندر شرط لکھی جاتی ہے۔ ایک شرط کوئی درست ایکپریشن ہو سکتی ہے جس میں ارجمند ایکپریشنز، ریلیشن ایکپریشنز، منطقی ایکپریشنز یا ان کا مجموعہ شامل ہو سکتا ہے۔

9: if شیئنٹ کے بنیادی بہاؤ کو ظاہر کرنے کے لیے ایک فلوچارٹ بنائیں۔



if-else: 10 if-else کو C لینگوچ میں کیوں استعمال کیا جاتا ہے؟

جواب: if - else if-else میں اگر مطلوب شرط درست ہو تو if شیئنٹ سے منسوب کردہ شیئنٹ کے سیٹ پر عمل درآمد ہوتا ہے ورنہ else شیئنٹ سے منسوب کردہ شیئنٹ کے سیٹ پر عمل درآمد ہوتا ہے۔

if-else: 11 if-else کا ذہانچہ بیان کریں؟

جواب: if-else شیئنٹ کا ذہانچہ یہ ہے:

(کندیش) if

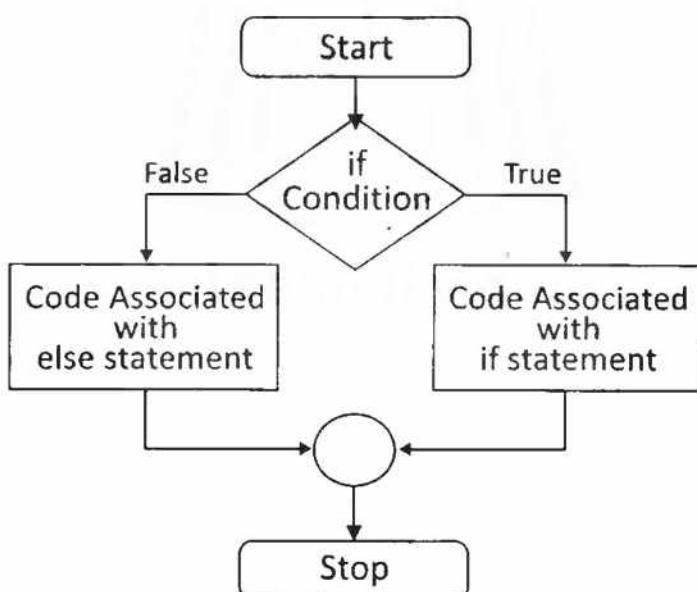
منسوب کردہ کوڈ

else

منسوب کردہ کوڈ

اگر کندیش درست ہے تو if شیئنٹ سے منسوب کردہ کوڈ پر عمل درآمد ہو گا ورنہ else شیئنٹ سے منسوب کردہ کوڈ پر عمل درآمد ہو گا۔

if-else: 12 شیئنٹ کا طویل بارٹ بنایں؟



13: کپاڈٹ ایٹمٹ کی تعریف کریں؟

جواب: بریکنوں میں بند متعدد ہدایات کا ایک مجموعہ بلاک یا کپاڈٹ ایٹمٹ کہلاتا ہے۔

14: if کے اندر اگر if-else ایٹمٹ ہے تو اسے آپ کیسے بند کر سکتے ہیں؟

جواب: اگر if ایٹمٹ یا else ایٹمٹ کے اندر ایک سے زیادہ ہدایات نہوں تو انھیں ایک بلاک (بیز می بریکنوں) کے اندر بند کریں۔ بصورت دیگر

کپاڈٹ مرف ایک ہدایت کو اس سے منسوب کر لے ہے اور باقی ہدایات کو ان سے الگ سمجھتا ہے۔

## مشروط منطق

110

15: نیشن سلیکشن سر کچر ز کیا ہے؟

جواب: کند بیش شیمنش کے اندر کند بیش شیمنش کو ہم نیشن سلیکشن سر کچر ز کرتے ہیں۔

16: نیشن سلیکشن سر کچر ز کا عمومی ڈھانچہ بیان کریں؟

جواب: درج ذیل سر کچر نیشن سلیکشن سر کچر کے لیے درست ہے۔

(پہلی کند بیش) if

(دوسری کند بیش) if

منسوب کردہ کوڈ

else

منسوب کردہ کوڈ

17: if-else-if شیمنٹ کا ڈھانچہ بیان کریں؟

جواب: C لیکوئج if-else-if شیمنٹ بھی فراہم کرتی ہے جس کا ڈھانچہ یہ ہوتا ہے۔

(پہلی کند بیش) if

کوڈ پر مولدر آمد کیا جائے اگر پہلی کند بیش درست ہے۔

(دوسری کند بیش) else if

کوڈ پر مولدر آمد کیا جائے گا اگر کند بیش 1 فلاٹ ہے لیکن کند بیش 2 درست ہے۔

N کند بیش (else if)

کوڈ پر مولدر آمد کیا جائے گا اگر تمام سابقہ کند بیش مطلوب ہیں لیکن کند بیش N درست ہے۔

else

یہ کوڈ بٹلے گا اگر تمام کند بیش مطلوب ہیں۔

18: ہم کپاڈٹا شیمنٹ میں کون سی عام غلطیاں کرتے ہیں؟

جواب: کپاڈٹا شیمنٹ میں ایک عام سی غلطی یہ ہے کہ ناچنگ کے درواں ایک یا دو بریکٹوں کو بھول جاتا ہے۔ اس ایرے سے پہنچ کے لیے بہتر ہے کہ پہلے اوپنگ اور کلوز گی بریکٹ ناچ کریں اور پھر بلاک میں شیمنٹ ناچ کریں (بیز می بریکٹوں میں بند)۔

19: نیشن سلیکشن سر کچر کا استعمال بیان کریں؟

جواب: نیشن سلیکشن سر کچری لیکوئج میں فیصلہ سازی کرنے کے لیے استعمال ہوتا ہے۔

20: کند بیش سے کیا مراد ہے؟

جواب: ایک کند بیش کوئی شرط یا درست ایکپر بیش ہو سکتی ہے جس میں ار تھیک ایکپر بیش، ریلیشن ایکپر بیش، مخفی ایکپر بیش یا ان کا مجموعہ شامل ہو سکتا ہے۔

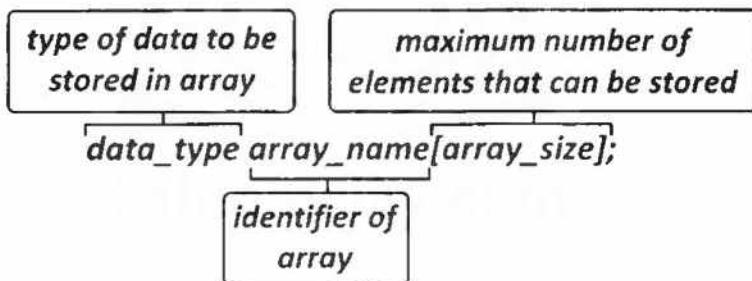
## پوئنٹ نمبر 4

### ڈیٹا اینڈ ریپیٹیشن

س: ڈیٹا سٹر کچھ کہا ہے؟ کون سا ڈیٹا سٹر کچھ زیادہ تر C پر وگر انگل لینگوچ میں استعمال ہوتا ہے؟  
 ڈیٹا سٹر کچھ ایک مخصوص ترتیب میں ڈیتا آئیز کے ذخیرہ کو محفوظ کرنے کا ایک کنٹریز ہے۔ C پر وگر انگل لینگوچ میں مختلف ڈیٹا سٹر کچھز دستیاب ہیں تاہم ایک ارے (Array) عام طور پر استعمال ہونے والے ڈیٹا سٹر کچھز میں سے ایک ہے۔  
 ارے (Array)

ارے ایک ڈیٹا سٹر کچھ ہے جو ایک ہی ڈیٹا ٹاپ کی ایک سے زیادہ قیمتیں رکھ سکتا ہے۔ ایک int ارے ایک سے زیادہ انٹیجر قیمتیں جبکہ فلوٹ (float) ارے ایک سے زیادہ تکلیفی قیمتیں رکھ سکتا ہے۔ ارے کی ایک اہم خصوصیت یہ بھی ہے کہ یہ تمام قیمتیں کو کمپیوٹر میموری کے اندر اکٹھا محفوظ کرتا ہے۔

س: ارے ڈیکلائریشن سے کیا مراد ہے؟ ارے ڈیکلائریشن کے تین حصوں کو مختصر طور پر بیان کریں۔  
 C لینگوچ میں ایک ارے کو مندرجہ ذیل طریقے سے ڈیکلائر کیا جاسکتا ہے۔



اگر ہم چاہتے ہیں کہ ایک انٹیجر ٹاپ کی ارے ہو جس میں ایک مزدور کی سات دن تک روزانہ کی اجرت رکھی جائے تو ہم اسے مندرجہ ذیل طریقے سے ڈیکلائر کر سکتے ہیں۔

int daily\_wage[7];

float marks[20];

درج ذیل مثال 20 طلباء کے نمبر رکھنے کے لیے فلوٹ ٹاپ ارے کی ڈیکلائریشن کی مثال ہے۔

س: ارے انٹیلائزریشن کی وضاحت مثال کی مدد سے کریں؟

چلی بار کسی ارے میں قیمتیں رکھنے کو ارے انٹیلائزریشن کہا جاتا ہے۔ ایک ارے کو اس کی ڈیکلائریشن کے وقت یا بعد میں انٹیلائزر کیا جاسکتا ہے۔ ڈیکلائریشن کے وقت ارے کی انٹیلائزریشن مندرجہ ذیل طریقے سے کی جاسکتی ہے۔

**Data\_type array\_name[N] = {value1, value2, value3,.....,valueN};**

مثال: درج ذیل مثال سات افراد کی قامت (height) کو ذخیرہ کرنے کے لیے فلوٹ ارے کی ڈیکلائریشن اور انٹیلائزریشن کی وضاحت کرتی ہے۔

float height[7] = {5.7, 6.2, 5.9, 6.1, 5.0, 5.5, 6.2};

مثال: ذیل میں ایک اور مثال ہے جو انگریزی زبان کے پانچ حروف علات (vowels) کو خیر کرنے کے لیے حروف کی ایک ارے کو انسٹالائز کیا گیا ہے۔

```
char vowels[5] = {'a', 'e', 'i', 'o', 'u'};
```

اگر ہم ڈیکھیریشن کے وقت ایک ارے کو انسٹالائز نہیں کرتے ہیں تو ہمیں ارے ایلینٹس کو ایک ایک کر کے انسٹالائز کرنے کی ضرورت ہے۔ اس کا مطلب یہ ہے کہ ہم ایک ہی شیئٹ میں ارے کے تمام ایلینٹس کو انسٹالائز نہیں کر سکتے۔ اس کی وضاحت ذیل کی مثال سے کی گئی ہے۔

void main() {     int array[5];     array[5] = {10, 20, 30, 40, 50}; }	ERROR: initialization whole array after declaration not allowed
--	---

کپیکل درج بالا مثال کے کوڈ پر ایک ایرر دیتا ہے جب ہم پوری ارے کو ڈیکھیریشن کرنے کے بعد ایک الگ شیئٹ میں انسٹالائز کرنے کی کوشش کرتے ہیں۔

سچ ہم ارے کے ایلینٹس تک کیسے رسائی حاصل کر سکتے ہیں؟ مختصر وضاحت کریں۔

ارے کے ہر ایلینٹ کا ایک انڈکس نمبر ہوتا ہے جسے ارے کے نام کے ساتھ اس طرح; array\_name[index]; لکھ کر ہم انڈکس میں مخصوص ایلینٹ کا رسائی حاصل کرنے کے لیے استعمال کر سکتے ہیں۔

پہلے ایلینٹ کا انڈکس 0 ہوتا ہے، دوسرے ایلینٹ کا انڈکس 1 ہوتا ہے اور آگے ایسے ہی چلتا ہے۔ لہذا [0] ارے height کے پہلے ایلینٹ کا حوالہ دیتا ہے۔ [1] سے مراد دوسرا ایلینٹ ہے اور اسی طرح آگے چلتا ہے۔ مثلاً

```
float height [7] = {5.7, 6.2, 5.9, 6.1, 5.0, 5.5, 6.2};
```

height[0]	height[1]	height[2]	height[3]	height[4]	height[5]	height[6]
Index	0	1	2	3	4	5
5.7	6.2	5.9	6.1	5.0	5.5	6.2
						Values

یہ اعداد دیکھارے height کی گرفیل نامہندی کو ظاہر کرتا ہے۔

س: ایک پروگرام لکھیں جو پانچ افراد کی عمر کو ایک ارے میں محفوظ کرے اور پھر سکرین پر دکھائے۔

**Program**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int age [5];
    age [0] = 25;
    age [1] = 34;
    age [2] = 29;
    age [3] = 43;
    age [4] = 19;
    printf("The ages of five persons are : %d, %d, %d, %d, %d", age [0], age [1], age [2], age [3], age [4]);
    getch();
}
```

س: ایک پروگرام لکھیں جو 4 مضمون میں حاصل کردہ نمبر صارف سے ان پٹ کے طور پر لے اور حاصل کردہ کل نمبر سکرین پر دکھائے۔

**Program**

```
#include<stdio.h>
#include<conio.h>
void main ()
{
    clrscr();
    float marks [4], total_marks;
    printf("Please enter the marks obtained in 4 subjects: ");
    scanf("%f%f%f%f", &marks[0], &marks[1], &marks[2], &marks[3]);
    total_marks = marks[0] + marks[1] + marks[2] + marks[3];
    printf ("Total marks obtained by student are %f ", total_marks);
    getch();
}
```

سوندھ کو ارے کے طور پر کیسے استعمال کیا جاسکتا ہے؟ مثال کی مدد سے وضاحت کریں۔

ارے کی ایک بہت اہم خصوصیت یہ ہے کہ ہم ویری ایتل کو ارے انڈس کس کے طور پر استعمال کر سکتے ہیں۔ درج ذیل پروگرام پر غور کریں:

#### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int array [5] = {10, 20, 30, 40, 50};
    int i;
    /* Following statements ask the user to input an index into variable i. */
    printf("Please enter the index whose value you want to display");
    scanf("%d", &i);
    /* Following statement displays the value of the array at the index entered by user. */
    printf("The value is %d", array[i]);
    getch();
}
```

جب ہم کسی ویری ایتل کی قیمت کو تبدیل کرتے ہیں تو اس کے بعد ویری ایتل نئی قیمت استعمال کرتا ہے۔

مثلاً:

#### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int array [5] = {10, 20, 30, 40, 50};
    int i = 2;
    /* Following statement displays value 30, as i contains 2 and the value at array[2] is 30 */
    printf("%d", array[i]);
    i++;
    /* Following statement displays value 40, as i has been incremented to 3 and the value at
array [3] is 40. */
    printf("\n%d", array[i]);
    getch();
}
```

س: لوپ سر کچھ کہا ہے؟ لوپس کے عمومی ذھانچے کو بیان کریں؟

اگر ہم ایک یا ایک سے زیادہ شیئٹس کو دہرانے کی ضرورت ہو تو ہم لوپس استعمال کرتے ہیں۔ مثال کے طور پر اگر ہمیں سکرین پر ہزار بار "لکھنے کی ضرورت ہے تو ہم ہزار بار printf("Pakistan"); لکھنے کی بجائے لوپس استعمال کریں گے۔ C لیٹکوچ تین قسم کے لوپ سر کچز فراہم کرتی ہے۔

1. For لوپ

2. While لوپ

3. Do-while لوپ

#### لوپ کا عام ذھانچہ (General structure of loops)

اگر ہم اس پروگرام کا پاریک بنی سے مشاہدہ کرتے ہیں جو اننان کسی کام کو مخصوص حد تک دہرانے کے لیے کرتا ہے تو ہمارے لیے لوپ سر کچز کو سمجھنا آسان ہو جاتا ہے جو کہ C لیٹکوچ ہمیں ہمارا کوئی نظرول کرنے کے لیے فراہم کرتی ہے۔

فرض کریں کہ ہمارے کھلیوں کے استاد ہم سے رنگ ٹریک کے 10 چکر لینے کو کہتے ہیں۔ ہم اس کام کو کیسے انجام دیں؟ پہلے ہم نے گنتی کو صفر کر دیا۔ کیونکہ ہم نے ابھی تک ٹریک کا ایک چکر بھی نہیں لگایا۔ پھر ہم چکر کا ناشروع کرتے ہیں۔ ہر راونڈ کے بعد ہم اپنی گنتی میں 1 کا اضافہ کرتے ہیں اور چکر کرتے ہیں کہ ہم نے 10 راونڈ مکمل کیے ہیں یا نہیں۔ اگر ہم نے ابھی تک 10 راونڈ مکمل نہیں کیے تو ہم پھر سے ایک راونڈ لیں گے اپنی گنتی میں 1 کا اضافہ کریں گے اور دوبارہ چکر کریں گے کہ ہم نے 10 راونڈ لیے ہیں یا نہیں۔ ہم اس عمل کو اس وقت تک دہراتے ہیں جب تک ہماری گنتی 10 تک نہ بہتچ جائے۔

ختلف پر گرامنگ لیٹکوچ میں بدایات کے ایک سیٹ کو دہرانے کے لیے لوپ سر کچز کے اسی فلسفے کی ہیرودی کی جاتی ہے۔

س: لوپ کے عام سینکڑس کی ٹھوپچارٹ اور مثال کے ساتھ وضاحت کریں۔

C پر گرامنگ لیٹکوچ میں for لوپ کے لیے درج ذیل سینکڑس ہے۔

**for (initialization; condition; increment / decrement) → For loop Statement**

{

**Code to repeat → Body of for loop**

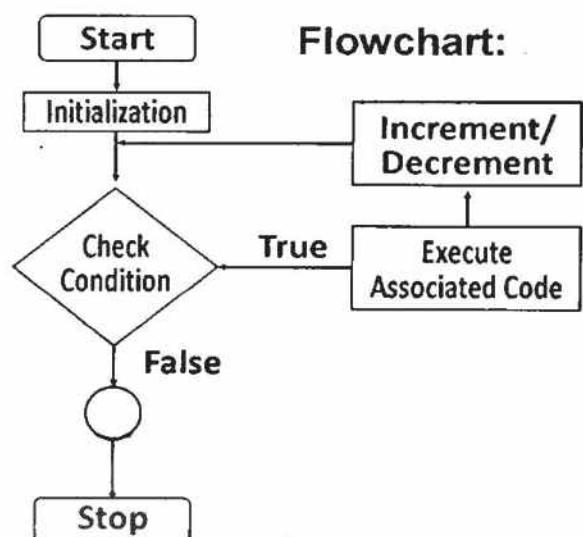
}

ٹھوپچارٹ سے ہم مندرجہ ذیل ترتیب کا مشاہدہ کر سکتے ہیں:

1. انشیلازیشن (initialization) for لوپ میں پہلا حصہ ہے جو سب سے پہلے ایگر کیوٹ ہوتا ہے۔ یہاں ہم اپنے کاؤنٹر وری ایبل کو انشیلازیشن کرتے ہیں اور پھر کندیشن والے حصے پر جاتے ہیں۔
2. کندیشن (Condition) چک کی جاتی ہے اور اگر یہ غلط (false) نہ ہے تو ہم لوپ سے باہر آ جاتے ہیں۔
3. اگر کندیشن درست (true) ہے تو لوپ کی باڑی چلتی ہے۔
4. لوپ کی باڑی ایگر کیوٹ ہونے کے بعد استعمال شدہ لا جک کے لحاظ سے کاؤنٹر وری ایبل کو بڑھایا یا کم کیا جاتا ہے اور پھر ہم دوبارہ مرحلہ 2 پر جاتے ہیں۔

مثال:

Program
#include<stdio.h> #include<conio.h> void main() { for(int i = 0; i < 3; i++) { printf ("Pakistan\n"); } }
Output
Pakistan Pakistan Pakistan



تفصیل:

اگر ہم تحریر کو ڈاکا مشاہدہ کرتے ہیں اور اس کا موازنہ فلوچارٹ کی تفصیل سے کرتے ہیں تو ہم ایکزکیوشن کی درج ذیل ترتیب دیکھ سکتے ہیں۔

1. ایشلائریشن ایکسپریشن چلتی ہے یعنی  $i = 0$ ; int میں کا اٹھوڑی ایتل "i" زیکریز ہو جاتا ہے اور قیمت 0 سے ایشلائز ہو جاتا ہے۔
2. کندیشن یعنی  $i < 3$  کو نیٹ کیا جاتا ہے جیسا کہ  $i$  کی قیمت 0 ہے جو 3 سے کم ہے لہذا کندیشن پوری ہو جاتی ہے اور ہم لوپ کی بادی میں آ جاتے ہیں۔
3. لوپ بادی چلتی ہے یعنی printf("Pakistan\n") اس طرح "Pakistan" سکرین پر ظاہر ہوتا ہے۔
4. انکریمینٹ / ڈیکریمینٹ ایکسپریشن ایکزکیوٹ ہوتی ہے یعنی  $i++$  اس طرح  $i$  کی قیمت میں 1 کا اضافہ کیا جاتا ہے جیسا کہ  $i$  کی قیمت پہلے 0 تھی لہذا اب قیمت 1 ہو جائے گی۔
5. اب کندیشن دوبارہ چک کریں گے کیونکہ  $i$  کی قیمت 1 ہے جو 3 سے کم ہے تو کندیشن دوبارہ درست ہو جاتی ہے اور لوپ بادی دوبارہ چلتی گی یعنی "Pakistan" دوبارہ سکرین پر دکھایا جاتا ہے۔ پھر  $i$  کی قیمت بڑھ کر 2 ہو جائے گی۔
6. اب دوبارہ کندیشن کو نیٹ کریں گے کیونکہ  $i$  کی قیمت 2 ہے جو 3 سے کم نہیں اس لیے کندیشن غلط (false) ہو جائے گی۔ اور کنٹرول لوپ سے باہر آجائے گا۔
7. اب دوبارہ کندیشن کا نیٹ کریں گے کیونکہ  $i$  کی قیمت 3 ہے جو 3 سے کم نہیں اس لیے کندیشن غلط (false) ہو جائے گی۔ اور کنٹرول لوپ سے باہر آجائے گا۔

س: لوپ کے لئے کا ڈنڑویری ا عمل کی وضاحت کریں؟

کا ڈنڑویری ا بیل ایک ویری ا بیل ہے جو کوڈ کے مخصوص حصے پر عملدرآمد کار بیکار ذرکرتا ہے کہ اس کوڈ کو کتنی بار دہرا یا جائے گا۔ for لوپ میں کا ڈنڑویری ا بیل اس لیے استعمال ہوتا ہے کہ ہر بار لوپ کے چلے سے قیمت میں کتنا اضافہ یا کمی ہوتی ہے۔

س: ایک پروگرام میں جو 1 سے 10 تک نمبر سکرین پر دکھائے۔

### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    for(int i = 1; i<= 10; i++)
    {
        printf("%d\n", i);
    }
    getch();
}
```

### تھیسیل:

درج بالا پروگرام پر غور کریں:

- سب سے پہلے i کی قیمت 1 رکھی اور پھر کندیشن چیک کی گئی ہے۔
- جیسا کہ کندیشن ( $10 == >$ ) درست (true) ہے تو لوپ باڑی عمل میں آتی ہے۔ جیسا کہ لوپ باڑی میں ہم کا ڈنڑویری ا بیل کی قیمت ظاہر کر رہے ہیں سکرین پر ظاہر ہوتا ہے۔
- اضافے کے بعد i کی قیمت 2 ہو جاتی ہے کندیشن کو دوبارہ چیک کیا جاتا ہے یہ درست (true) ہے کیونکہ ( $10 == > 2$ ) ہے اس لیے 2 پر نٹ کیا گیا ہے۔
- یہ عمل جاری رہتا ہے جب تک کہ 10 ظاہرنہ ہو جائے اور اضافے کے بعد i کی قیمت 11 ہو جائے۔ اب کندیشن غلط (false) ہو گئی کیونکہ ( $11 > 10$ ) سے اب لوپ آخر میں 1 سے 10 تک نمبر پر نٹ کرنے کے بعد لوپ ختم ہو جائے گا۔

ہمیشہ اس بات کو تلقینی بنائیں کہ کندیشن آگے کمیں جا کر غلط ہو جائے ورنہ لوپ چلتا رہے گا کبھی ختم نہیں ہو گا۔

### تکرار (Iteration)

تکرار وہ عمل ہے جہاں بدایات یا بیانات کا مجموعہ ایک مخصوص تعداد تک یا کندیشن کے غلط (false) ہونے تک بار بار چلا یا جاتا ہے۔

س: ایک پروگرام لکھیں جو صرف سے ایک نمبر ان پہلے اور اس کا فیکٹریل شد کرے۔  
 پروگرام کی منطق (Logic of the Program): جب ہم کسی مسئلے کو پروگرام کے مطابق حل کرنا چاہتے ہیں تو ہم سے پہلے ہمیں یہ جانتے کی ضرورت ہوتی ہے کہ ہم کیا حاصل کرنا چاہتے ہیں۔ اس مثال میں ہمیں کسی دیے گئے نمبر کا فیکٹریل معلوم کرنے کی ضرورت ہے لہذا پہلے ہمیں نمبر کا فیکٹریل معلوم کرنے کے لیے فارمولہ جانے کی ضرورت ہے۔

$$N! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots \cdot (N-1) \cdot N$$

ہم اس پیرن کو دیکھ سکتے ہیں جو درج ایجاد ہے لہذا ہم لوپ کا استعمال کرتے ہوئے مسئلہ کو حل کر سکتے ہیں۔

### Program

```
#include<stdio.h>
#include<conio.h>

void main()
{
    clrscr();
    int n, fact = 1;
    printf("Please enter a positive number whose factorial you want to find");
    scanf("%d", &n);
    for(int i = 1; i<= n; i++)
    {
        fact = fact * i;
    }
    printf("The factorial of input number %d is %d", n, fact);
    getch();
}
```

تفصیل: مندرجہ ذیل نیل پروگرام کے کام کو عاہر کرتا ہے اگر ان پڑھو۔ اس میں ہر ٹکرار کے بعد وہی ای بلز کی قیمتیں میں تبدیلی واضح کی گئی ہے۔

Iteration	Value of counter	Condition	Loop body	Result
				fact=1
1	i=1	TRUE(1<=5)	fact = fact * i	fact=1*1=1
2	i=2	TRUE(2<=5)	fact = fact * i	fact=1*2=2
3	i=3	TRUE(3<=5)	fact = fact * i	fact=2*3=6
4	i=4	TRUE(4<=5)	fact = fact * i	fact=6*4=24
5	i=5	TRUE(5<=5)	fact = fact*i	fact=24*5=120
6	i=6	FALSE (6>5)		120 is shown on output screen

س: نیشنل لوب کاٹھانچ بیان کریں؟ ہم نیشنل لوب کیوں استعمال کرتے ہیں؟  
ایک لوپ کے اندر دوسرا لوپ نیشنل لوب کہلاتا ہے۔ for لوپ کا عمومی سڑکمر یہ ہے۔

```
for (initialization; condition; increment / decrement)
{
    Code to repeat
}
```

ہم مشاہدہ کر سکتے ہیں کہ دہرایا جانے والا کوڈ (code to repeat) C لینگوچ کا کوئی بھی کوڈ ہو سکتا ہے۔ یہ ایک اور for لوپ بھی ہو سکتا ہے۔  
درج ذیل سڑکمر ایک درست نیشنل لوب سڑکمر ہے۔

```
for (initialization; condition; increment / decrement)
{
    for (initialization; condition; increment / decrement)
    {
        Code to repeat
    }
}
```

### نیشنل لوب کا استعمال (Use of nested loop):

جب ہم ایک لوپ کو دوسرے لوپ کے اندر استعمال کرتے ہیں تو اسے نیشنل لوب سڑکمر کہا جاتا ہے۔ جب ہم ایک بیرون کو کئی بار دہراناچاہتے ہیں تو ہم نیشنل لوب استعمال کرتے ہیں جیسے اگر 10 بار ہم 1 سے 10 تک نمبر دکھاناچاہتے ہیں تو ہم یہ کر سکتے ہیں کہ 1 سے 10 نمبروں کو ظاہر کرنے کا کوڈ کسی اور لوپ میں لکھیں جو 10 بار چلتا ہے۔

س: ایک پروگرام لکھیں جو 5 مرتبہ کمپیوٹر کی سکرین پر 10-1 تک نمبر دکھائے۔

### Program

```
#include<stdio.h>
#include<conio.h>
void main ()
{
    clrscr();
    for(int i = 1; i<= 5; i++)
    {
        for(int j = 1; j <= 10; j++)
        {
            printf("%d ", j);
        }
        print("\n");
    }
    getch();
}
```

### Output

```
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
```

### تفصیل

جیسا کہ ہم اندر ونی لوپ کے کام کو سمجھتے ہیں تو یہاں ہم تبیر ونی لوپ پر غور کریں گے۔

1. n کی قیمت 1 ہو تو تبیر ونی لوپ میں کندیشن چیک کی جاتی ہے جو کہ درست ہے ( $5 \geq 1$ ) لہذا اپور اندر ونی لوپ چلتا ہے اور 1 سے 10 تک نمبر سکرین پر پرنٹ کر دیتا ہے۔

2. جب کنٹرول اندر ونی لوپ سے باہر آتا ہے تو ;(" ")printf چلتا ہے اور سکرین پر ایک خالی لائن آ جاتی ہے۔

3. پھر n کی قیمت میں اضافہ ہوتا ہے اور یہ 2 ہو جاتی ہے۔ جیسا کہ یہ 5 سے کم ہے لہذا کندیشن درست ہے۔ لہذا اپور اندر ونی لوپ چلتی ہے اور اس طرح 1 سے 10 تک نمبر دوبارہ سکرین پر پرنٹ ہو جاتے ہیں۔ پھر اندر ونی لوپ سے باہر آ کر ایک خالی لائن کسکوں پر ڈال دی جاتی ہے۔

4. سکرین پر 1 سے 10 تک نمبروں کو 5 بار ظاہر کرنے کے بعد n کی قیمت بڑھ کر 6 ہو جاتی ہے اور تبیر ونی لوپ کی کندیشن غلط ہو جاتی ہے۔ تبیر ونی لوپ بھی ختم ہو جاتا ہے۔

س: سکرین پر ۔ کے درج ذیل پہنچن کو ظاہر کرنے کے لیے ایک پروگرام لکھیں۔

```
*
**
***
****
*****
******
*****
```

### Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    for(int i = 1 ; i <= 6; i++)
    {
        for(int j = 1 ; j <= i; j++)
            printf ("*");
        printf ("\n");
    }
    getch();
}
```

### Output

```
*
**
***
****
*****
******
*****
```

### تفصیل

- مندرجہ بالا کوڈ کی تفصیل یہ ہے۔
- جیسا کہ ہم ستاروں کی 6 لاکھیں پر نٹ کرنا پاچتے ہیں۔ اس کی بیردی لوپ کو 1 سے 6 تک چلاتے ہیں۔
  - ہم دیکھ سکتے ہیں کہ دیئے گئے پہنچن میں ہمارے پاس پہلی لائی پر 1 ستارہ، دوسری لائی پر 2 ستارے، تیسرا لائی پر 3 ستارے اور اسی طرح باقی ہیں۔ لہذا اندر ونی لوپ کا کاؤنٹر 1 ہے تو اندر ونی لوپ ابار چلتے گا۔ اگر بیردی لوپ کا کاؤنٹر 2 ہے تو اندر ونی لوپ 2 بار چلتے گا اور اسی طرح باقی۔ لہذا ہم بیردی لوپ کا کاؤنٹر اندر ونی لوپ کی کندھیں میں استعمال کرتے ہیں یعنی  $i \leftarrow j$
  - جب بیردی لوپ کا کاؤنٹر کی قیمت 1 ہوتی ہے اندر ونی لوپ صرف 1 بار چلتا ہے تو صرف 1 ستارہ پر نٹ ہوتا ہے۔ جب بیردی لوپ کا کاؤنٹر 2 ہوتا ہے تو اندر ونی لوپ 2 بار چلتا ہے لہذا 2 ستارے پر نٹ ہوتے ہیں اور اس عمل کو دہرا یا جاتا ہے جب تک کہ 6 لاکھیں مکمل نہ ہو جائیں۔
  - ہم if سڑک پر زکو لوپ سڑک پر میں اور لوپ سڑک پر کو if سڑک پر میں کسی بھی قابل ذکر طریقے سے استعمال کر سکتے ہے۔

س: ایک پروگرام لکھیں جو بتائے کہ دونوں نمبروں کے درمیان دیئے گئے نمبر کے کتنے مالک ضرب آتے ہیں۔

**Program**

```
#include <stdio.h>
#include<conio.h>
void main ()
{
    clrscr();
    int n, lower, upper, count = 0;
    printf ("Enter the number: ");
    scanf ("%d", &n);
    printf ("Enter the lower and upper limit of multiples:\n");
    scanf ("%d%d", &lower, &upper);
    for(int i = lower; i<= upper; i++)
        if(i % n == 0)
            count++;
    printf ("Number of multiples of %d between %d and %d are %d", n, lower, upper, count);
    getch();
}
```

س: ایک پروگرام لکھیں جو n1 سے n2 تک آنے والے اشیاء میں سے جتنے نمبراتے۔ (n1 & n2 میں سے)

**Program**

```
#include <stdio.h>
#include<conio.h>
void main ()
{
    clrscr();
    int n1, n2;
    printf ("Enter the lower and upper limit of even numbers:\n");
    scanf ("%d%d", &n2, &n1);
    if(n1>n2)
    {
        for(int i = n1; i >= n2; i--)
        {
            if(i% 2 ==0)
                printf("%d", i );
        }
    }
    getch();
}
```

س: ایک پروگرام لکھیں جو یہ بتائے کہ دیا گیا نمبر مفرد (prime) نمبر ہے یا نہیں۔

**Program**

```
#include <stdio.h>
#include<conio.h>
void main ()
{
    clrscr();
    int n;
    int flag = 1;
    printf ("Enter a number: ");
    scanf ("%d", &n);
    for (int i = 2; i < n; i++)
    {
        if (n% i == 0)
            flag = 0;
    }
    if (flag == 1)
        printf ("This is a prime number");
    else
        printf ("This is not a prime number");
    getch();
}
```

س: ایک پروگرام لکھیں جو 2 سے 100 تک کے مفرد اعداد سکرین پر دکھائے۔

**Program**

```
# include<stdio.h>
#include<conio.h>
int main ()
{
    clrscr();
    int flag;
    for (int j = 2; j <= 100; j++)
    {
        flag = 1;
        for (int i = 2; i < j; i++)
        {
            if (j %i ==0)
            {
                flag = 0;
            }
        }
        if (flag ==1)
        {
            printf ("%d ", j);
        }
    }
    getch();
}
```

س: لوپس اور اریز میں کا فرق ہے؟ ارے میں قیتوں کو پڑھنے اور لکھنے کے لیے لوپس کو کس طرح استعمال کیا جاسکتا ہے۔ جیسا کہ دیری ایبل کو ارے اندیکس کے طور پر استعمال کیا جاسکتا ہے لہذا ہم اریز پر مختلف آپریشن کرنے کے لیے لوپس استعمال کر سکتے ہیں۔ اگر ہم پوری ارے کو پرمنٹ کرنا چاہتے ہیں تو پھر تمام ایمیٹیشن کو ایک ایک کر کے لکھنے کی بجائے ہم لوپ کا ڈنٹ کو ارے اندیکس کے طور پر استعمال کرتے ہوئے ارے پر لوپ چلا سکتے ہیں۔ ارے میں قیتوں کو پڑھنے اور لکھنے کے طریقہ درج ذیل ہیں۔

### 1) لوپس کا استعمال کرتے ہوئے ارے میں ٹیکسٹ لکھنے:

لوپس کا استعمال کرتے ہوئے ہم آسانی سے ارے میں ان پڑے لے سکتے ہیں۔ اگر ہم 10 سائز کی ارے میں صارف سے ان پڑے لینا چاہتے ہیں تو ہم لوپ کو درج ذیل طور پر استعمال کر سکتے ہیں۔

```
int a [10];
for (int i = 0; i < 10; i++)
scanf ("%d", &a[i]);
```

س: ایک پروگرام لکھیں جو 5 سائز کی ارے میں 23 کے پہلے 5 حاصل ضرب لکھے۔

#### Program

```
#include<stdio.h>
void main()
{
    int multiples[5];
    for (int i = 0; i < 5; i++)
        multiples[i] = (i + 1)*23;
}
```

### 2) لوپس کا استعمال کرتے ہوئے ارے سے ٹیکسٹ پڑھنا:

ارے سے قیتوں کو پڑھنے میں لوپس ہماری مدد کرتے ہیں۔ مندرجہ ذیل کوڈ کو استعمال کر کے ہم 100 ایمیٹیشن کی ایک ارے سے سکرین پر دکھائے سکتے ہیں۔

```
for (int i = 0; i < 100; i++)
printf("%d ", a[i]);
```

درج ذیل کوڈ کو استعمال کر کے ہم 100 ایمیٹیشن کی ایک ارے کے تمام ایمیٹیشن کو جمع کر سکتے ہیں۔

```
int sum = 0;
for(int i = 0; i < 100; i++)
    sum = sum + a[i];
printf("The sum of all the elements of array is %d", sum);
```

س: ایک پروگرام لکھیں جو دو اریز کے مختلف ایمیشنز مچ کرے۔

### Program

```
#include <stdio.h>
void main ()
{
    int a [] = { 2, 3, 54, 22, 67, 34, 29, 19};
    int b [] = {65, 73, 26, 10, 4, 2, 84, 26};
    for (int i = 0; i<8; i++)
        printf ("%d", a[i] + b[i]);
}
```

### مشق

### س 1: بھیر الاتھابی سوالات

نمبر	سوال	A	B	C	D
1	اے ایک _____ سڑک ہے۔	لوپ	کنڑول	ذیٹا	مشروط
2	اے کے ایمیشنز میوری کے مقامات _____ پر محفوظ ہوتے ہیں۔	فائل	بکھرے ہوئے	تقسیم شدہ	کوئی بھی نہیں
3	اگر ایک اے کا سائز 100 ہے تو انڈکس کی ریخ _____ ہو گی۔	0-99	0-100	1-100	2-102
4	سڑک ہمیشہ ہدایات کے ایک سیٹ کو بار بار دہرانے کے لیے استعمال ہوتا ہے۔	لوپ	مشروط	کنڑول	ذیٹا
5	ایک مخصوص شاخت ہے جو اے کا خواہ دہتا ہے۔	ذیٹا کی قسم	اے کا نام	اے کا سائز	کوئی بھی نہیں
6	اے کو ذیکلائیزیریشن کے _____ انسٹالائر کیا جاتا ہے۔	اس وقت	اٹکے بعد	اس کے پہلو	اور بدو نوں la
7	لوپ کے اندر لوپس کا استعمال کو _____ لوپس کہا جاتا ہے۔	For	While	Do-while	Nested
8	لوپ کا _____ حصہ سب سے پہلے چلتا ہے۔	کنڑشن	بازی	انڈکسائزرن	اضافہ / کی

س: ایک پروگرام لکھیں جو دو اریز کے مختلف ایمیشنز مچ کرے۔

### Program

```
#include <stdio.h>
void main ()
{
    int a [] = { 2, 3, 54, 22, 67, 34, 29, 19};
    int b [] = {65, 73, 26, 10, 4, 2, 84, 26};
    for (int i = 0; i<8; i++)
        printf ("%d", a[i] + b[i]);
}
```

### مشق

### س 1: بھیر الاتھابی سوالات

نمبر	سوال	A	B	C	D
1	اے ایک _____ سڑک ہے۔	لوپ	کنڑول	ذیٹا	مشروط
2	اے کے ایمیشنز میوری کے مقامات _____ پر محفوظ ہوتے ہیں۔	فائل	بکھرے ہوئے	تقسیم شدہ	کوئی بھی نہیں
3	اگر ایک اے کا سائز 100 ہے تو انڈکس کی ریخ _____ ہو گی۔	0-99	0-100	1-100	2-102
4	سڑک ہمیشہ ہدایات کے ایک سیٹ کو بار بار دہرانے کے لیے استعمال ہوتا ہے۔	لوپ	مشروط	کنڑول	ذیٹا
5	ایک مخصوص شاخت ہے جو اے کا خواہ دہتا ہے۔	ذیٹا کی قسم	اے کا نام	اے کا سائز	کوئی بھی نہیں
6	اے کو ذیکلائیزیریشن کے _____ انسٹالائر کیا جاتا ہے۔	اس وقت	اٹکے بعد	اس کے پہلو	اور بدو نوں la
7	لوپ کے اندر لوپ کا استعمال کو _____ لوپ کہا جاتا ہے۔	For	While	Do-while	Nested
8	لوپ کا _____ حصہ سب سے پہلے چلتا ہے۔	کنڑشن	بازی	انڈیکسیشن	اضافہ / کی

نکشن	اکپریشن	کنٹرول	لوپس	سے ارے میں قیمتی لکھنا اور پڑھنا آسان ہو جاتا ہے۔	9
اور دونوں la	پہلے	بعد	وقت	ایک ہی سینکڑ میں ارے کو انسٹالائز کرنے کے لیے اسے ذیکریشن کے انسٹالائز کریں۔	10

جوابات

1	2	3	4	5	6	7	8	9	10
C	A	A	A	B	A	D	C	A	A

س2: درج زیل سوالوں کی وضاحت کریں۔

ملٹ نمبر 111 پر لاحظہ فرمائیں

(1) ڈیتا سٹر کچر

ملٹ نمبر 111 پر لاحظہ فرمائیں

(2) ارے

ملٹ نمبر 111 پر لاحظہ فرمائیں

(3) ارے انسٹالائزیشن

ملٹ نمبر 115 پر لاحظہ فرمائیں

(4) لوپ سٹر کچر

(5) نیشنیڈ لوپس

س3: مندرجہ زیل سوالات کے مختصر جوابات دیں۔

(1) کہا لوپ ایک ڈیتا سٹر کچر ہے؟ اپنے جواب کی توثیق کریں۔

جواب: لوپ ڈیتا سٹر کچر نہیں ہے کیونکہ ڈیتا سٹر کچر کا مطلب کپیور میوری کی ایک مخصوص ترتیب ہے جس میں ڈیتا آئنسز کا ذخیرہ محفوظ کیا جائے جبکہ لوپ سٹیشن کے بیٹ کو دہرانے کے لیے استعمال ہوتا ہے۔

ملٹ نمبر 119 پر لاحظہ فرمائیں

(2) نیشنیڈ لوپس کا استعمال کیا ہے؟

(3) ایک ارے کو ذیکریشن کے وقت انسٹالائز کرنے کا کام کہا ہے؟

جواب: ذیکریشن کے وقت ایک ارے انسٹالائز کرنے کے درج زیل فوائد ہیں۔

وقت کی بچت (Save Time)

اگر آپ رن نام پر ارے کو انسٹالائز کرتے ہیں تو یہ وقت بچانے میں مدد کرتا ہے جائے اس کے کہ آپ ارے کو خود داخل کریں یا سٹیشن کے ساتھ scanf

میوری کی بچت (Save Memory)

انسٹالائز ارے خود کو داس کے ڈیتا کے ساتھ رن نام پر میوری مخفی کرتی ہے۔ اس سے پروگرام کی کارکردگی بہتر ہوتی ہے۔

CPU فرینڈلی (CPU Friendly)

ایک بار اے انیشلائز ہونے کے بعد اس ارے پر آپریشن سر انجام دینے کے لیے CPU آسانی سے رسائی حاصل کر سکتا ہے۔ ہم ارے پر بہت تیز تکرار انجام دے سکتے ہیں۔

**سائز کی وضاحت (Define the Size):**

ارے کو انیشلائز کرتے ہوئے آپ کے پاس دو مختلف اختیاب ہوتے ہیں۔ ارے [n] اور ارے []۔ ارے [n] میں آپ ارے کے سائز کی وضاحت کر سکتے ہیں اور اگر آپ ارے کا سائز نہیں جانتے تو آپ ارے [] استعمال کر سکتے ہیں یہ آپ کو ارے کے نامعلوم سائز پر کام کرنے میں مدد دیتا ہے۔

4) for لوپ کی ساخت بیان کریں؟

5) آپ ارے کو کیسے ڈیلکٹریز کر سکتے ہیں؟ ارے ڈیلکٹریز کے عنوان حصوں کو مختلف طور پر بیان کریں۔

6: درج ذیل کوڈ کے حصوں میں لفظیوں کی شناخت کریں۔

a) int a[] = {{2},{3},{4}};

**Error:** Parenthesis is not used in array initialization and single curly braces are used for 1d array.

b) for (int i = 0, i< 10, i++)

    printf("%d\n", i);

**Error:** Semicolon is missing inside the loop.

c) int a[ ] = {1,2,3,4,5};

    for (int j = 0; j< 5; j + +)  
        printf ("%d ", a(j));

**Error:** Square bracket is used instead of parenthesis in printf statement.

d) float f[] = (1.4, 3.5, 7.3, 5.9);

    int size = 4;  
    for (int n = 1; n < size; n--)  
        printf ("%f\n", f[n]);

**Error:** Loop condition never be false.

e) int count = 0;

    for(int i=4; i < 6; i -- )  
        for(int j=i; j < 45; j + +)  
            {  
                count++;  
                printf("%count",count)  
            }

**Error:** Outer loop never be false. Semicolon is missing at the end of printf statement.

س5: درج ذیل کوڈ کے حصول کی آوث پٹ لکھیں۔

a) int sum = 0, p;  
 for (p = 5; p <= 25; p = p + 5)  
 sum = sum + 5;  
 printf ("sum is %d ", sum);

**Output:** 25

b) int i;  
 for (i = 34; i<= 60; i \* 2)  
 printf("\*");

**Output:** \*

c) for (int i = 50; i<= 50; i++)  
 {  
 for (j = i; j >= 48, j - -)  
 printf ("j=%d \n", j);  
 printf ("i = %d\n", i);  
 }

**Output:**  
 j=50  
 j=49  
 j=48  
 i=50

d) int i, arr[ ] = {2, 3, 4, 5, 6, 7, 8};  
 for (i = 0; i <7; i++)  
 {  
 printf ("%d\n" arr[i] \* arr[i]);  
 i++;  
 }

**Output:**  
 4  
 16  
 36  
 64

e) int i, j;  
 float ar1[ ] = {1.1, 1.2, 1.3};  
 float ar2[ ] = {2.1, 2.2, 2.3};  
 for (i = 0; i < 3; i ++)  
 for (j = i ; j < 3; j++)  
 printf("%f\n", ar1[i] \* ar2[j] \* i \* j)

**Output:**  
 0.000000  
 0.000000  
 0.000000  
 2.640000  
 5.520000  
 11.959999

## پروگرام کی مشق

خانہ 1:

.....  
.....  
.....

1) لوپس کو استعمال کر کے کنسول پر پھر ان پرنٹ کریں۔

### Program

```
#include <stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    int i, j;
    for(i=1;i<=3;i++)
    {
        for(j=1;j<=5;j++)
            printf("*");
        printf("\n");
    }
    getch();
}
```

### Output

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

A  
BC  
DEF  
GHIJ  
KLMN

2) لوپس کو استعمال کر کے کنسول پر پھر ان پرنٹ کریں۔

### Program

```
#include <stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    char ch='A';
    for(int i=1;i<=5;i++)
    {
        for(int j=1;j<=i; j++)
            printf("%c",ch++);
        printf("\n");
    }
    getch();
}
```

### Output

A  
BC  
DEF  
GHIJ  
KLMN

## ڈیٹا اینڈ ریپیشیشن

130

مشن 2: ایک پروگرام لکھیں جو دو نوبت مدد نہ اور  $a^b$  کو بلور ان پٹ لیتا ہے اور  $a^b$  شد کر کے سکرین پر دکھائے۔

### Program

```
#include <math.h>
#include <stdio.h>
#include <conio.h>

int main()
{
    clrscr();
    int a, b, result;
    printf("Enter a base number: ");
    scanf("%d", &a);
    printf("Enter an exponent: ");
    scanf("%d", &b);
    result = pow(a, b);
    printf("%d^%d = %d", a, b, result);
    getch ();
}
```

### Output

```
Enter a base number: 2
Enter an exponent: 3
2^3=8
```

مشن 3: ایک پروگرام لکھیں جو دو نمبر ان پٹ کے طور پر لیتا ہے اور یوں کلیدین طریقہ استعمال کرتے ہوئے (GCD) معلوم کر کے بتاتے۔

### Program

```
#include<stdio.h>
#include<conio.h>
int main()
{
    clrscr();
    int m, n;
    printf("Enter two integer numbers: ");
    scanf ("%d \n %d", &m, &n);
    while (n > 0)
    {
        int r = m % n;
        m = n;
        n = r;
    }
    printf ("GCD = %d \n",m);
    getch ();
}
```

### Output

```
Enter two integer numbers: 5
15
GCD = 5
```

## ڈیٹا اینڈ ریپیٹیشن

131

**مشن 4:** ایک پروگرام لکھیں جو 1 سے 7 تک نمبر کا فیکریل سکرین پر دکھائے۔ (اشارہ: نیشنل لوبس استعمال کریں)

**Solution:**

```
#include <stdio.h>
#include <conio.h>
int main()
{
    clrscr();
    int i;
    unsigned long int fact = 1;
    for(int n=1;n<=7;n++)
    {
        for(i = n; i> 0; i--)
        {
            fact = fact * i;
        }
        printf("%d \t",fact);
        fact=1;
    }
    getch();
}
```

**Output**

1	2	6	24	120	720	5040
---	---	---	----	-----	-----	------

**مشن 5:** ایک پروگرام لکھیں جو ایک 10 ایمیٹنس کی ایک ارے کو ٹیکلیئر اور انشا کر کرے اور پہلے اور آخری ایمیٹس کو ضرب کر کے جواب سکرین پر دکھائے۔

**Program**

```
#include <stdio.h>
#include <conio.h>
int main()
{
    clrscr();
    int num[10];
    printf("Enter 10 Numbers :");
    for(int i =0;i<10;i++)
    {
        scanf("%d\n",&num[i]);
    }
    printf("Product of 1st and last number is =%d",num[0]*num[9]);
    getch ();
}
```

**Output**

Enter 10 Numbers : 9
8
7
9
6
5
26
6
5
4

Product of 1st  
and last number is = 36

## ڈیٹا اینڈ ریپیشیشن

132

مشن 6: ایک پروگرام لکھیں جو 7 ایٹمیں کی ایک ارے ڈیکسٹر اور انہیں تحریر کرے اور یہ بتائے کہ ارے کے کتنے ایٹمیں 10 سے بڑے ہیں۔

### Program

```
#include <stdio.h>
#include <conio.h>
int main()
{
    clrscr();
    int num[7]={9,8,11,12,14,15,16},count=0;
    for(int i =0;i<7;i++)
    {
        if(num[i]>10)
            count++;
    }
    printf("The Elements greater than 10 in array are :%d",count);
    getch();
}
```

### Output

The Elements greater than 10 in array are : 5

## سرگرمیوں کا حل

سرگرمی 4.1: ایک پروگرام لکھیں جو 2 کا نتیجہ پر منٹ کرے۔

### Program

```
#include <stdio.h>
#include <conio.h>
int main()
{
    clrscr();
    int n=2, i;
    for (i = 1; i<= 10; ++i)
    {
        printf("%d * %d = %d \n", n, i, n * i);
    }
    getch ();
}
```

### Output

2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20

مرگی 4.2: ایک پروگرام کھیس جو ایک کلاس کے 30 طلباء کے میٹرک میں حاصل کردہ نمبروں کو ان پڑ کے طور پر لے۔ اور ان کی اوسط بتائے۔

**Program**

```
#include <stdio.h>
#include <conio.h>
int main()
{
    clrscr();
    int marks[30],total=0;
    float avg;
    printf("Enter Marks of 30 students \n");
    for(int i=0;i<30;i++)
    {
        scanf("%d\n",&marks[i]);
        total=total+marks[i];
    }
    avg = total/30;
    printf("Average Marks is =%f ",avg);
    getch ();
}
```

**Output**

**Note:** All values are entered in consecutive 30 lines, the table refer only example.

Enter Marks of 30 students

515	600	450	750	850	477	985	1052	1014	914
814	715	625	525	423	498	478	1011	615	698
874	748	987	789	879	654	546	546	564	456

Average Marks is = 701.733333

مرکزی 4.3: ایک پروگرام لکھیں جو 2, 3, 4, 5 اور 6 کے نتیجے پر نت کرے۔

**Program****Solution:**

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n=2;
    for (int i = 1; i<= 10; ++i)
    {
        printf("%d * %d = %d \n", n, i, n * i);
    }
    printf("\n\n");
    n=3;
    for (int i = 1; i<= 10; ++i)
    {
        printf("%d * %d = %d \n", n, i, n * i);
    }
    printf("\n\n");
    n=4;
    for (int i = 1; i<= 10; ++i)
    {
        printf("%d * %d = %d \n", n, i, n * i);
    }
    printf("\n\n");
    n=5;
    for (int i = 1; i<= 10; ++i)
    {
        printf("%d * %d = %d \n", n, i, n * i);
    }
    printf("\n\n");
    n=6;
    for (int i = 1; i<= 10; ++i)
    {
        printf("%d * %d = %d \n", n, i, n * i);
    }
    getch ();
}
```

**Output:**

2 * 1 = 2	5 * 1 = 5
2 * 2 = 4	5 * 2 = 10
2 * 3 = 6	5 * 3 = 15
2 * 4 = 8	5 * 4 = 20
2 * 5 = 10	5 * 5 = 25
2 * 6 = 12	5 * 6 = 30
2 * 7 = 14	5 * 7 = 35
2 * 8 = 16	5 * 8 = 40
2 * 9 = 18	5 * 9 = 45
2 * 10 = 20	5 * 10 = 50
3 * 1 = 3	6 * 1 = 6
3 * 2 = 6	6 * 2 = 12
3 * 3 = 9	6 * 3 = 18
3 * 4 = 12	6 * 4 = 24
3 * 5 = 15	6 * 5 = 30
3 * 6 = 18	6 * 6 = 36
3 * 7 = 21	6 * 7 = 42
3 * 8 = 24	6 * 8 = 48
3 * 9 = 27	6 * 9 = 54
3 * 10 = 30	6 * 10 = 60
4 * 1 = 4	
4 * 2 = 8	
4 * 3 = 12	
4 * 4 = 16	
4 * 5 = 20	
4 * 6 = 24	
4 * 7 = 28	
4 * 8 = 32	
4 * 9 = 36	
4 * 10 = 40	

**Output**

Note: Table should display below to next table and vice versa.

## کشیر الاتجاحی سوالات

ارے	ڈیتا سٹر کچورز	ہارڈ دیزئر	سافت دیزئر	ایک مخصوص ترتیب میں ڈینا آئنڈر کے ذخیرہ کو محفوظ کرنے کا ایک کشیر ہے۔	1
اگر تم	فلوچارت	ارے	لوپ	ایک سب سے زیادہ استعمال ہونے والے ڈینا سٹر کچورز میں سے ایک ہے۔	2
کوئی بھی نہیں	لوپ	سیوری	ارے	ایک ڈینا سٹر کچورہ جو ایک قیڈی ڈینا ناچہ کی ایک سے زیادہ قیties رکھ سکتا ہے۔	3
int daily_wage [[7]];	int daily_wage (7);	int daily_wage [7];	int daily_wage [7]	ایک اشیعیر ناچہ کی ارے ہو جس میں ایک مزدور کی سات دن تک روزانہ کی اجرت رکھی جائے تو ہم اسے طریقے سے ذیکر کر سکتے ہیں۔	4
2	1	0	-1	ایک ارے انڈیکس سے شروع ہوتا ہے۔	5
کوئی بھی نہیں	35	36	9	اگر ایک int سائز 4 باسٹس کا ہو تو ارے [9] کا سائز کیا ہو گا؟	6
int arr(2) = (10, 20)	int arr[2] = {10, 20};	int arr(2)= {10, 20}	int arr[2]= (10, 20)	ہم C لینگوچ میں ایک ارے کو کیسے انشیلاز کر سکتے ہیں۔	7
ارے انشیلازریشن	ارے ذیکریزیشن	ارے فائلزیشن	ارے فلٹ	پہلی بار کسی ارے میں قیties رکھنا کہلاتا ہے۔	8
کوئی بھی نہیں	لنسٹ	انشیلازریشن	ذیکریزیشن	ایک ارے کو کے وقت انشیلازر کیا جاسکتا ہے۔	9
ایک ساتھ	تم تم کر کے	دودو کر کے	ایک ایک کر کے	اگر ہم ذیکریزیشن کے وقت ایک ارے کو انشیلازر نہیں کرتے تو ہم ارے ایلمیشن کو انشیلازر کرنے کی ضرورت ہوتی ہے؟	10
nested	float	int	while	ایک ارے متعدد اشیعیر قیties رکھ سکتا ہے۔	11
while	float	Simple	for	ایک ارے متعدد حقیقی قیties رکھ سکتا ہے۔	12
کوئی بھی نہیں	کوئی بھی نہیں	مختلف مقامات پر محفوظ	ایک قیمت کو محفوظ	ارے کی ایک اہم خصوصیت یہ بھی ہے کہ کے تمام قیتوں کو کپیوڑ سیوری کے اندر کرتا ہے۔	13

ارے کے نام	ارے لوگیشن	لوپ	انڈکس	ارے کے ہر الیمنٹ کا ایک انڈکس نمبر ہوتا ہے جو ----- کے ساتھ اس طرح array_name[index]; لکھ کر ہم انڈکس میں مخواہ دیاں اسکے لیے شامل کر سکتے ہیں۔	14
شیبر	کانٹٹ	ویری ایبل	نمبر	ارے کی ایک بہت ابھم خصوصیت یہ ہے کہ ہم ----- کو ارے انڈکس کے طور پر استعمال کر سکتے ہیں۔	15
یہ سب	لوپس	ٹکرار	ارے	اگر ہمیں ایک یا ایک سے زیادہ شیٹھنٹس کو دھرانے کی ضرورت ہو تو ہم ----- استعمال کرتے ہیں۔	16
6	5	4	3	C لینگوچ قسم کے لوپ سڑکچرز میا کرتی ہے۔	17
لوپ check	لوپ while	لوپ do while	لوپ for	مندرجہ ذیل میں سے کون سی لوپس کی قسم نہیں ہے؟	18
کوئی بھی نہیں	فیلائریشن	انشاہریشن	ڈیکلائریشن	لوپ میں پہلا حصہ ہے جو سب سے پہلے ایک ریکیوٹ ہوتا ہے۔	19
رنگ	ٹکرار	ریپیٹیشن	ڈیکلائریشن	ہر بار لوپ کے چلنے کو کہا جاتا ہے۔	20
do while loop	while loop	nested loop	else loop	جب ہم ایک لوپ کو دوسرا لوپ کے اندر استعمال کرتے ہیں تو اسے سڑکچرز کہا جاتا ہے۔	21
nested loop	else loop	do while loop	repetition loop	جب ہم ایک پیڑن کو کئی بار دھرانا چاہتے ہیں تو ہم ----- استعمال کرتے ہیں۔	22
نیشنل سڑکچر	while	تسلیل سڑکچر	لوپ سڑکچر	ہم لوپ سڑکچرز کے اندر ----- استعمال کر سکتے ہیں۔	23
if سڑکچر	loop	سڑکچر data	سڑکچر while	اگر سڑکچر تصوراتی آداب میں ہوں تو ہم ----- استعمال کر سکتے ہیں۔	24
کانٹٹ	ڈیتا سڑکچر	لوپ	ویری ایبل	بلوراے انڈکس کے طور پر استعمال کیا جاسکتا ہے۔	25
ڈیٹا	ارے	لوپس	انفارمیشن	ہم ----- پر مختلف آپریشن کرنے کے لیے لوپس استعمال کر سکتے ہیں۔	26
نیشنل لوپ	ارے	لوپس	لوپ for	----- کا استعمال کرتے ہوئے ہم آسانی سے ارے میں ان پت لے سکتے ہیں۔	27

کوئی بھی نہیں	کپاٹر	ارے	لوپس	ارے سے ٹیکسٹ کو پڑھنے میں ہماری مدد کرتی ہے۔	28
کوئی بھی نہیں	for (انیشلائریشن کندیشن; اگر یعنیت / ذکر یعنیت)	for (اگر یعنیت / ذکر یعنیت; انیشلائریشن; کندیشن)	for (انیشلائریشن; کندیشن ؛ اگر یعنیت / ذکر یعنیت)	لوب کے لیے درست سینکڑ کیا ہے	for
رن ٹائم ایر	کپاٹر ایر	ہاں	نہیں	کیا for لوب کے اندر کوئی دوسری for لوب ہو سکی ہے۔	30

جوابات:

1	2	3	4	5	6.	7	8	9	10
C	B	A	B	B	B	C	D	A	A
11	12	13	14	15	16	17	18	19	20
B	C	A	D	B	C	A	D	A	C
21	22	23	24	25	26	27	28	29	30
B	D	A	C	A	B	B	A	A	B

## غیر سوالات

س 1. ڈینا سٹر کچر کی وضاحت کریں۔

جواب: ڈینا سٹر کچر ایک مخصوص ترتیب میں ڈینا آئیمز کے ذخیرہ کو محفوظ کرنے کا ایک سینیز ہے۔

س 2. ارے کیا ہے؟

جواب: ارے ایک ڈینا سٹر کچر ہے جو ایک ڈینا ٹاپ کی ایک سے زیادہ ٹیکسٹ رکھ سکتا ہے اور تمام قیتوں کو کپیوریٹ میوری کے اندر اکٹھا محفوظ کرتا ہے۔

س 3. ہم int ٹاپ کی ارے کی ڈیکلائریشن کیسے کر سکتے ہیں؟

جواب: اگر ہم چاہتے ہیں کہ ایک انشیجر ٹاپ کی ارے ہو جس میں ایک مزدور کی مسات دن تک روزانہ کی اجرت رکھی جائے تو ہم اسے مندرجہ ذیل طریقے سے ڈیکلائر کر سکتے ہیں۔

س 4. ہم فلوت ٹاپ ارے کی ڈیکلائریشن کیسے کر سکتے ہیں؟

جواب: 20 طلباء کے نمبر رکھنے کے لیے فلوت ٹاپ ارے کی ڈیکلائریشن کی مثال ہے۔

س 5. ارے کی انیشلائریشن سے کیا مراد ہے؟

جواب: پہلی بار کسی ارے میں ٹیکسٹ رکھنے کو ارے انیشلائریشن کہا جاتا ہے۔ ایک ارے کو اس کی ڈیکلائریشن کے وقت یا بعد میں انیشلائر کیا جا سکتا ہے۔

س. 6. ہم ارے کو کیسے انیشلائر کر سکتے ہیں؟

جواب: ڈیکلائریشن کے وقت ارے کی انیشلائزیشن مدرجہ ذیل طریقے سے کی جاسکتی ہے۔

```
Data_type array_name[N] = {value1, value2, value3,.....,valueN};
```

س. 7. سات افراد کی قام (height) کو ذخیرہ کرنے کے لیے قوٹ ارے کی ڈیکلائریشن اور انیشلائزیشن کی مثال لکھیں۔

```
float height[7] = {5.7, 6.2, 5.9, 6.1, 5.0, 5.5, 6.2};
```

جواب:

س. 8. کا آپ ارے کا سائز تائے بغیر ایک ارے کی ڈیکلائریشن کر سکتے ہیں؟

جواب: نہیں، ہم ارے کے سائز کا تین کی بغیر ارے کی ڈیکلائریشن نہیں کر سکتے۔

س. 9. اگر ری لینگوچ کے پانچ حروف ملت (vowels) کو ذخیرہ کرنے کے لیے ارے انیشلائزیشن کی مثال لکھیں۔

```
char vowels[5] = {'a', 'e', 'i', 'o', 'u'};
```

جواب:

س. 10. اگر ہم ڈیکلائریشن کے وقت انیشلائزیشن کرتے ہیں تو ہم ارے کو کیسے انیشلائر کر سکتے ہیں؟

جواب: اگر ہم ڈیکلائریشن کے وقت کسی ارے کو انیشلائزیشن کرتے ہیں تو ہم ارے کا یونیٹس کو ایک ایک کر کے انیشلائز کرنے کی ضرورت ہے۔ اس کا مطلب یہ ہے کہ ہم ایک ہی شیئٹ میں ارے کے تمام یونیٹس کو انیشلائزیشن کر سکتے۔

س. 11. کیا ہم ایک ہی شیئٹ میں ارے کے تمام یونیٹس کو انیشلائز کر سکتے ہیں؟ اسے مثال کے ساتھ بیان کریں۔

جواب: نہیں، ہم ایک ہی شیئٹ میں ارے کے تمام یونیٹس کو انیشلائزیشن کر سکتے۔ ایر درج ذیل مثال میں غایب ہوتا ہے:

<pre>void main() {     int array[5];     array[5] = {10, 20, 30, 40, 50}; }</pre>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">ERROR : initialization whole array after declaration not allowed</div>
---	---

س. 12. ہم ارے یونیٹس ہج کیسے رسمی حاصل کر سکتے ہیں؟

جواب: ارے کے ہر یونیٹ کا ایک انڈسکس نمبر ہوتا ہے جسے ارے کے نام کے ساتھ اس طرح `array_name[index]` لکھ کر ہم انڈسکس میں محفوظ ذیانا ہج رسمی حاصل کرنے کے لیے استعمال کر سکتے ہیں۔

س. 13. ویری ایبلز کو ارے کے طور پر کیسے استعمال کا جا سکتا ہے؟

جواب: ارے کی ایک بہت اہم خصوصیت یہ ہے کہ ہم ویری ایبل کو ارے انڈسکس کے طور پر استعمال کر سکتے ہیں۔ درج ذیل پروگرام پر غور کریں:

```
#include<stdio.h>
void main()
{
    int array [5] = {10, 20, 30, 40, 50};
    int i;
    printf("Please enter the index whose value you want to display");
    scanf("%d", &i);
    printf("The value is %d", array[i]);
}
```

س 14. لوپ کیا ہے؟

جواب: اگر ہم ایک یا ایک سے زیادہ شیشنس کو دیرانے کی ضرورت ہو تو ہم لوپ کی استعمال کرتے ہیں۔ مثال کے طور پر اگر ہمیں سکرین پر پڑا بار "Pakistan" لکھنے کی ضرورت ہے تو ہم بار بار ("Pakistan") printf لکھنے کے جایے لوپ کی استعمال کریں گے۔

س 15. C لیٹکون گھنی لوپ کے کون سے سڑ کچڑ فراہم کیے گئے ہیں؟

جواب: C لیٹکون گھنی حتم کے لوپ سڑ کچڑ فراہم کرتی ہے۔

لوپ Do-while .3

لوپ While .2

لوپ For .1

س 16. لوپ کا ممکنہ ذھانیہ بیان کریں؟

جواب: اگر ہم اس پرو سس کا بار بار نبی سے مشاہدہ کرتے ہیں جو انسان کسی کام کو مخصوص حد تک دیرانے کے لیے کرتا ہے تو ہمارے لیے لوپ سڑ کچڑ کو سمجھنا آسان ہو جاتا ہے جو کہ C لیٹکون گھنی میں بھر اک کنڑول کرنے کے لیے فراہم کرتی ہے۔

س 17. لوپ کے ممکنہ ذھانیہ کا سنیک لکھیں؟

جواب: C پر دوستیک لیٹکون گھنی میں for لوپ کے لیے درج ذیل سنیک ہے۔

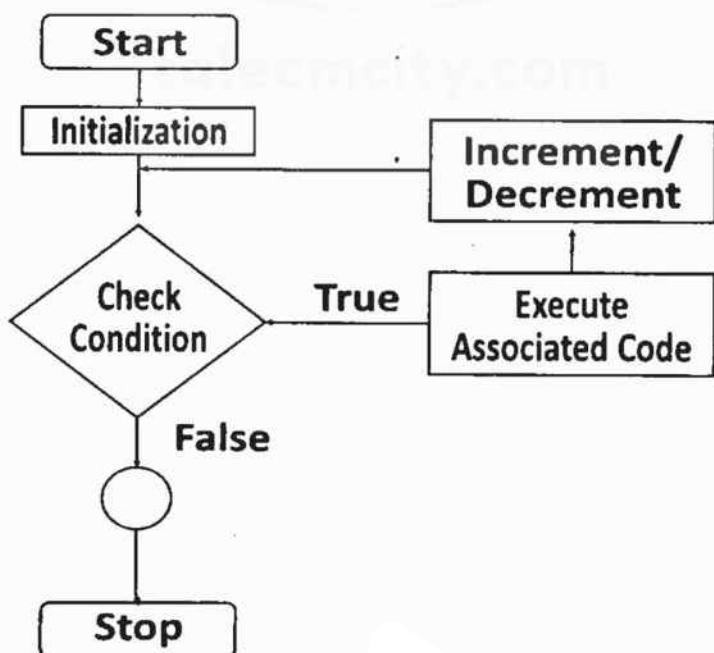
```
for (initialization; condition; increment / decrement)
```

{

Code to repeat

}

س 18. لوپ سڑ کچڑ کے لیے ٹکوچارٹ بنائے۔



س 19. لوب کے کس حصے کو پہلے ایگر کیوٹ کیا جاتا ہے۔

جواب: اینیشلائزیشن (initialization) لوب میں پہلا حصہ ہے جو سب سے پہلے ایگر کیوٹ ہوتا ہے۔ یہاں ہم اپنے کام نویری ایجل کو اینیشلائز کرتے ہیں اور پھر کنٹرول وانے حصے پر جاتے ہیں۔

س 20. درج زمیں پر گرام کی آٹھ پٹ کیا ہو گی؟

```
int main()
{
    int i = 0, x = 0;
    do
    {
        if(i % 5 == 0)
        {
            cout<<x;
            x++;
        }
        ++i;
    }
    while(i<10);
    cout<<x;
    getch ();
}
```

**Output**  
012

س 21. درج زمیں پر گرام کی آٹھ پٹ کیا ہو گی؟

```
int main()
{
    int i=0,x=0;
    for(i=1;i<10;i+=2)
    {
        x++;
        cout<<x;
    }
    cout<<x;
    getch ();
}
output  
12344
```

س 22. ہمار کی تعریف کریں؟

جواب: ہمارا وہ عمل ہے جہاں بدایاں یا بیانات کا مجموعہ ایک مخصوص عدد اسکے کائنڈیشن کے خلاف (false) ہونے تک بار بار چلا جاتا ہے۔

س 23. نیشنیڈ لوپس کا بنیادی فناخچہ ہیاں کریں؟

جواب: ہم مشاہدہ کر سکتے ہیں کہ دھرا جانے والا کوڈ (code to repeat) C لیگریج کا کوئی بھی کوڈ ہو سکتا ہے۔ یہ ایک اور for لوپ بھی ہو سکتا ہے۔ درج ذیل سڑکم ایک درست نیشنیڈ لوپ سڑکم ہے۔

**for (initialization; condition; increment/decrement)**

{

**for (Initialization; condition; Increment/decrement)**

{

**Code to repeat**

}

}

س 24. نیشنیڈ لوپ سڑکم ہے؟

جواب: جب ہم ایک لوپ کے اندر دوسری لوپ کو استعمال کرتے ہیں تو اسے نیشنیڈ لوپ سڑکم کہا جاتا ہے۔

س 25. ہم نیشنیڈ لوپس کب استعمال کرتے ہیں؟

جواب: جب ہم ایک لوپ کو دوسرے لوپ کے اندر استعمال کرتے ہیں تو اسے نیشنیڈ لوپ سڑکم کہا جاتا ہے۔ جب ہم ایک پیڑن کو کئی بار دھرا جائیں تو ہم نیشنیڈ لوپس استعمال کرتے ہیں جیسے اگر 10 بار ہم 1 سے 10 تک نمبر دھانا چاہتے ہیں تو ہم یہ کر سکتے ہیں کہ 1 سے 10 نمبروں کو ظاہر کرنے کا کوڈ کسی اور لوپ میں لکھیں جو ہم اب بار چلتے ہے۔

س 26. لوپ اور ارے میں کیا فرق ہے؟

جواب:

ارے	لوپ
اے ایک ڈیٹا سڑکم ہے جو ایک ہی ڈیٹا ناچ کی ایک سے زیادہ قیمتیں رکھ سکتا ہے اور تمام قیمتیں کو کپیور میوری کے اندر اکٹھا محفوظ کرتا ہے۔	اگر ہم ایک یا ایک سے زیادہ سٹیٹمنٹس کو دھرانے کی ضرورت ہو تو ہم لوپس استعمال کرتے ہیں۔ مثال کے طور پر اگر ہم سکرین پر بزرگ بزرگ "Pakistan" لکھنے کی ضرورت ہے تو ہم بزرگ بزرگ ("Pakistan") printf لکھنے کے بعدے لوپس استعمال کریں گے۔

س 27. while لوپ کو کتنی بار پرنٹ کیا جاتا ہے؟

```
int main()
{
    int i = 1 ;
    i = i - 1 ;
    while(i)
    {
        cout<<"its a while loop";
        i++ ;
    }
    getch ();
}
Output
Infinite Times.
```

س 28. اگر ہم 10 سارے کی ارے میں صارف سے ان پڑ لیتا چاہتے ہیں تو ہم لوپ کا استعمال کیسے کر سکتے ہیں؟

جواب:

```
int a [10];
for (int i = 0; i< 10; i++)
scanf ("%d", &a[i]);
```

س 29. 100 ایمیٹر پر مشتمل ایک ارے کے ایمیٹر کو پرنٹ کرنے کے لیے کوڈ لکھیں۔  
جواب: درج ذیل کوڈ کو 100 ایمیٹر والے ارے کے ایمیٹر کو پرنٹ کرنے کے لیے استعمال کیا جاسکتا ہے۔

```
for (int i = 0; i< 100; i++)
printf("%d ", a[i]);
```

س 30. 100 ایمیٹر پر مشتمل ارے کے تمام ایمیٹر کو جمع کرنے کے لیے کوڈ لکھیں۔  
ج: درج ذیل کوڈ کو استعمال کر کہ ہم 100 ایمیٹر کی ایک ارے کے تمام ایمیٹر کو جمع کر سکتے ہیں۔

```
int sum = 0;
for(int i = 0; i< 100; i++)
sum = sum + a[i];
printf("The sum of all the elements of array is %d", sum);
```

س 31. While لوب کی مثال لکھیں۔

```
#include <stdio.h>
int main()
{
    int count=1;
    while (count <= 4)
    {
        printf("%d ", count);
        count++;
    }
    getch();
}
Output:
1 2 3 4
```

جواب:

س 32. for لوب کے لیے کا وظیر ویری ایتل کی وضاحت کریں؟

جواب: کا وظیر ویری ایتل ہے جو کوڈ کے خصوصی حصے پر عملدرآمد کاریکاری ذرکرتا ہے کہ اس کو ذکر کرنی بارہ براہ ایجاد کئے گا۔

س 33. for لوب میں کا وظیر ویری ایتل کے استعمال کی وضاحت کریں؟

جواب: for لوب میں کا وظیر ویری ایتل اس لیے استعمال ہوتا ہے کہ ہر بار لوب کے چلنے سے قیمت میں کتنا اضافہ یا کمی ہوتی ہے۔

س 34. کام لوب ایک ڈائیاٹر کچھ ہے؟ اپنے جواب کا جواز پیش کریں۔

جواب: لوب ڈائیاٹر کچھ نہیں ہے کیونکہ ڈائیاٹر کچھ زکا مطلب کپیوریٹ میوری کی ایک خصوصی ترتیب ہے تاکہ ڈینا آئیز کا ذخیرہ محفوظ کیا جاسکے جبکہ لوب شیئنر کے سیٹ کو دہرانے کے لیے استعمال ہوتا ہے۔

س 35. While لوب کی وضاحت کریں؟

جواب: پروگرامنگ لینگوچیں میں while لوب بار بار یہ ناتیا یادیات کا ایک سیٹ چلاتا ہے جب تک کہ دی گئی کنڈیشن درست ہو۔

س 36. While لوب کی ساخت بیان کریں؟

جواب:

```
while (condition)
{
    Statement or set of statements;
}
```

س 37. ذیکریزیشن کے وقت ارے انسٹالائزیشن کا کیا فائدہ ہے؟

ج: ذیکریزیشن کے وقت ایک ارے انسٹالائز کرنے کے درج ذیل فوائد ہیں:

1. وقت کی بچت      2. میوری کی بچت      3. فریغذی      4. سائز کی وضاحت

## یونٹ نمبر 5

### فناشز

س: آپ مسئلہ حل کرنے کے طریقہ کار اور تقسیم کرو اور فتح کرو کی تکنیک کے بارے میں کیا جاتے ہیں؟

مشکل اور چیزیں سائل کو حل کرنے کا عمل مسئلے کا حل کہلاتا ہے۔ کسی مسئلے کو حل کرنے کے لیے ایک منظم حکمت عملی اختیار کرنا ہوتی ہے۔ تقسیم کرو اور فتح کرو کی حکمت عملی میں ایک چیزیں سائل کو چھوٹے چھوٹے سائل میں تقسیم کیا جاتا ہے اور ان کا حل نکال کر ایک چیزیں سائل کا حل لیا جاتا ہے۔ اس طرح ہمارے لیے ایک وقت میں ایک چھوٹے سائل پر توجہ مرکوز کرنا آسان ہو جاتا ہے، مجھے اس کے کہ ہم ہر وقت پورے سائل کے بارے میں سوچیں۔

س. فناشز کیا ہے؟ فناشز کی اقسام کی وضاحت کریں۔

فناشن شیئننس کا ایک بلکہ ہے جو ایک خاص کام انجام دیتا ہے جیسے printf ایک فناشن ہے جو کمپیوٹر اسکرین پر کسی بھی چیز کو ظاہر کرنے کے لیے استعمال ہوتا ہے۔ ایک اور فناشن ہے جو یوں زر سے ان پت لینے کے لیے استعمال ہوتا ہے۔ ہر پروگرام میں ایک میں فناشن ہوتا ہے جو یوں زر کے کاموں کو سرانجام دیتا ہے۔ اسی طرح ہم دوسرے فناشز لکھ سکتے ہیں اور انہیں کمپیوٹر پر اسکے لئے بار استعمال کر سکتے ہیں۔

فناشز کی اقسام

بنیادی طور پر فناشز کی دو قسم ہیں۔

(1) بُلٹ ان فناشز (User-defined Function)      (2) یوڈیفائائنڈ فناشز (Built-in Function)

(1) بُلٹ ان فناشز (Built-in Function)

جن فناشز کی پہلے سے ہی پروگرام لیٹکنگ میں وضاحت کی جاتی ہے انہیں بُلٹ ان فناشن کہا جاتا ہے۔ ان فناشز کو لا بصری فناشن بھی کہا جاتا ہے۔ یہ فناشز لیٹکنگ کے طور پر دستیاب ہیں۔ یہ فناشز ریڈی میڈی پروگرام ہیں۔ یہ فناشز عام طور پر ریاضی کے حساب کتاب، سڑنگ آپریشن، ان پت/آؤٹ پت آپریشن وغیرہ انجام دیتے ہیں۔ مثال کے طور پر printf اور scanf بُلٹ ان فناشز ہیں۔

(2) یوڈیفائائنڈ فناشز (User-defined Function)

وہ فناشز جو پروگرام بناتا ہے انہیں یوڈیفائائنڈ فناشز کہتے ہیں۔ یہ فناشز یوں زر کی ضرورت کے مطابق بنائے جاتے ہیں۔

س. فناشز استعمال کرنے کے فوائد بیان کریں۔

فناشز ہمیں درج ذیل فوائد فراہم کرتے ہیں۔

(1) دوبارہ استعمال (Reusability)

فناشز کے ذریعے ہم کو دوبارہ استعمال کر سکتے ہیں۔ اس کا مطلب یہ ہے کہ جب بھی ہمیں ایک فناشن کی فناشیلیٹی (functionality) کی ضرورت ہو، ہم اس فناشن کو کال کر سکتے ہیں۔ ہمیں بار بار شیئننس کا ایک ہی مجموعہ لکھنے کی ضرورت نہیں ہے۔

بھر جیسیں اس مسئلے کو دور کرنے کے لیے پورے پروگرام کو چیک کرنے کی ضرورت نہیں ہے۔ ہمیں صرف ایک فناکشن پر توجہ دینے کی ضرورت ہے۔

### (3) مسئلے کی چیزیں کی سے ممٹا (Handling the Complexity of problem)

اگر ہم پورے پروگرام کو ایک پر دیسپر کے طور پر لکھیں تو پروگرام کی دیکھ بھال کرنا مشکل ہو جاتا ہے۔ فناکشن پر دیگرام کو چھوٹے چھوٹے حصوں میں تقسیم کرتے ہیں اور اس طرح مسئلے کی چیزیں کم ہو جاتی ہے۔

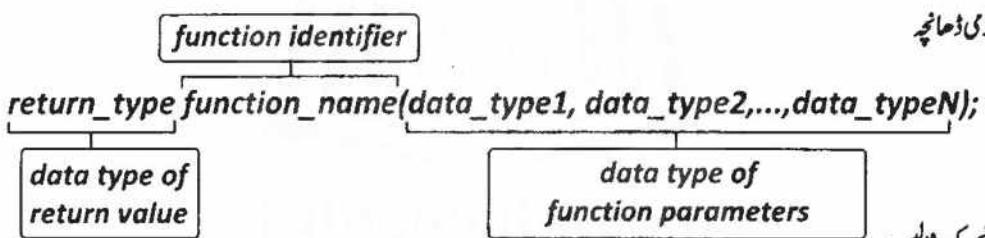
### (4) پڑھنے کی الیت (Readability)

پروگرام کو کئی فناکشن میں تقسیم کرنا پر دیگرام کی پڑھنے کی الیت کو بہتر بناتا ہے۔

س۔ فناکشن کا سینچر اور اس کا عمومی ذھانچہ مثالوں کے ساتھ بیان کریں۔

فناکشن سینچنٹس کا ایک بلاک ہے جو کچھ ان پت حاصل کرتا ہے اور کچھ آئٹ پت فراہم کرتا ہے۔ فناکشن کے ہمراہ اسے زکر جاتا ہے اور فناکشن کی آئٹ پت کو اس کی ریزن ویلو کہا جاتا ہے۔ ایک فناکشن میں ایک سے زیادہ ہمراہ اسے زیادہ قیمت ریزن نہیں کر سکتے۔ فناکشن کے سینچر (signature) فناکشن کی ان پت اور آئٹ پت کی وضاحت کے لیے استعمال ہوتے ہیں۔

فناکشن کا عمومی ذھانچہ



فناکشن سینچر کی مثالیں:

فناکشن سینچر	فناکشن کی تفصیل
int square (int);	ایک فناکشن جو ایک اسٹیجر کو بطور ان پت لیتا ہے اور اس کا مرعن ریزن کرے گا۔
float perimeter (float, float);	ایک فناکشن جو مستطیل کی لمبائی اور چوڑائی کو بطور ان پت لیتا ہے اور مستطیل کا احاطہ ریزن کرے گا۔
int largest (int, int, int);	ایک فناکشن جو تین اسٹیجر کو بطور ان پت لیتا ہے اور ان میں سب سے بڑی قیمت ریزن کرے گا۔
float area (float);	ایک فناکشن جو دائرے کے رادس کو بطور ان پت لیتا ہے اور دائرے کا رقبہ ریزن کرے گا۔
int isVowel (char);	ایک فناکشن جو ایک کیٹر کو بطور ان پت لیتا ہے اور اگر حرف علات (vowel) ہے تو ریزن کرے گا اور نہ 0 ریزن کرے گا۔

س: فونکشن ڈیٹینیشن کے حصوں کی تعریف کریں؟

ایک فونکشن کے سلیقہ سے یہ نہیں پڑھا کہ وہ کس کام کے لیے ہے۔ اس کے لیے فونکشن ڈیٹینیشن ہوتی ہے۔ ایک فونکشن ڈیٹینیشن ہا ڈھانچہ کچھ اس طرح سے ہے۔

```
return_type function_name (data_type_var1, data_type_var2, ...., data_type_varN)
{
    Body of the function
}
```

فونکشن کی باڑی ان شیئنیشن کا سیٹ ہوتا ہے جنہیں چلا کر فونکشن ایک خاص کام سرانجام دیتا ہے۔ فونکشن سلیقہ کے فوراً بعد تو سین ( ) کے اندر بند شیئنیشن کا مجموعہ فونکشن کی باڑی بناتا ہے۔

مثال: درج ذیل ایک فونکشن ( ) کی وضاحت کرتی ہے جو کوئی ان پڑھنے لیتا اور کچھ بھی ریزن نہیں کرتا اس کپیوٹر سکرین پر "A Quick Brown Fox Jumps over the Lazy Dog "

```
void showPangram()
{
    printf("\n A quick brown fox jumps over the lazy dog.\n");
}
```

جیسا کہ مذکورہ فونکشن کچھ ریزن نہیں کرتا اس طرح فونکشن کی ریزن تاپ void ہے۔

مثال: فونکشن کی مثال جو بطور ان پڑھ دو شیئر لیتی ہے اور دونوں شیئر کا مجموعہ ریزن کرے گی۔

```
int add (int x, int y)
{
    int result;
    result = x + y;
    return result;
}
```

فونکشن کے اندر return ایک مطلوب لفظ ہے جو کالنگ (calling) فونکشن میں وظیور ریزن کرنے کے لیے استعمال ہوتا ہے۔ ایک فونکشن ایک سے زیادہ قیمتیں ریزن نہیں کر سکتا۔ مثال کے طور پر مندرجہ ذیل سینٹس ایک کپاٹر ایر رکا نتھ ہے۔

Return (4,5);

ایک فونکشن میں ایک سے زیادہ ریزن شیئنیشن ہو سکتے ہیں لیکن جیسے ہی پہلی ریزن سینٹ چلتی ہے فونکشن کا ل ختم ہو جاتی ہے اور فونکشن کی باڑی میں مرید شیئنیشن پر عملدرآمد نہیں ہوتا ہے۔

س: فونکشن کال کرنے کے لیے کون ساموئی ڈھانچہ استعمال کیا جاتا ہے؟

بمیں کسی فونکشن کو کال کرنے کی ضرورت ہے تاکہ وہ پروگرام کو سونپا گیا کام انجام دے۔

عمومی ڈھانچہ (General Structure)

مندرجہ ذیل عمومی ڈھانچہ ہے جو فونکشن کال کرنے کے لیے استعمال ہوتا ہے۔

```
function_name(value1, value2, ..., valueN);
```

```
#include<stdio.h>
#include<conio.h>
void ShowPangram()
{
    printf("A quick brown fox jumps over the lazy dog\n");
}
void main()
{
    clrscr();
    printf("Hello from main()\n");
    ShowPangram();           ← function call
    printf("Welcome back to main()");
    getch();
}
```

### Output

```
Hello from main()
A quick brown fox jumps over the lazy dog.
Welcome back to main()
```

ہم دیکھ سکتے ہیں کہ پروگرام main() فونکشن سے چلتا شروع کرتا ہے۔ جب یہ ایک فونکشن کال (ستھیل کے اندر) آتی ہے تو یہ کنڑوں اس فونکشن میں منتقل ہو جاتا ہے۔ فونکشن کی سٹیشنس پر عمل کرنے کے بعد کنڑوں ریئن کائیں فونکشن میں منتقل ہو جاتا ہے۔

س: ایک پروگرام لکھیں جو دو نمبر انہٹے اور ان کا مجموعہ دکھائے۔  
 سٹیشن 1: sum = add (n1, n2); کو دیں فونکشن "add" کو کال کر لیں۔

```
void main()
{
    int n1, n2, sum;
    scanf ("%d%d", &n1, &n2);
    sum = add (n1 , n2);           ← Function arguments
    printf ("Sum is %d", sum);
}
```

فونکشن کاں میں n1 اور n2 فونکشن () add کے آر گو منس ہیں۔ فونکشن add() سے ریٹن آنے والے نتائج کو ذخیرہ کرنے کے لیے مخفی sum ڈیکلر کیا گیا ہے۔

فونکشن کو بطور آر گو منس پاس کیے گئے ویری ایبلز میں کوئی جدید نہیں آتی۔ فونکشن ویری ایبلز کی ایک کاپی بناتا ہے اور تمام تر ایم سرف اس کاپی میں کی جاتی ہیں۔ درج ذیل مثال میں جب n1 اور n2 پاس کیے جاتے ہیں تو فونکشن ان ویری ایبلز کی کاپیاں بناتا ہے۔ ویری ایبل n1 کی کاپی x ہے اور ویری ایبل n2 کی کاپی y ہے۔

سوال: آر گو منس اور ہی ایم سرف میں کیا فرق ہے؟ ایک مثال دیں۔

جو قیمتیں فونکشن کو پاس کی جاتی ہیں وہ آر گو منس کہلاتی ہے جبکہ فونکشن ڈیفینیشن میں جس ویری ایبلز میں یہ قیمتیں جاتی ہیں انہیں فونکشن کے ہی ایم سرف کہا جاتا ہے۔

مثال: sum = add (n1, n2);

اس مثال میں ویری ایبلز n1 اور n2 آر گو منس ہیں جو فونکشن () add کو پاس کیے ہوئے ہیں۔ جبکہ فونکشن () add کے اندر ویری ایبلز x اور y فونکشن کے ہی ایم سرف ہیں۔

س: کیا فونکشن ڈیفینیشن اور فونکشن کاں میں ایک جگہ ذیل ٹاپ استعمال کرنا ضروری ہے؟ اپنے جواب کو ایک مثال کے ساتھ بیان کریں۔  
یہ ضروری نہیں ہے کہ ویری ایبلز کو انہیں ناموں کے ساتھ فونکشن میں کاں کیا جائے جیسا کہ ہی ایم سرف کے نام ہیں۔ تاہم ہم ایک جیسے نام بھی استعمال کر سکتے ہیں۔ یہاں ایک اور اہم نکتہ یہ بھی ہے کہ اگر ہم ایک جیسے نام استعمال کرتے ہیں جب بھی فونکشن میں استعمال ہونے والے ویری ایبلز اصل ویری ایبلز کی ایک کاپی ہوں گے۔ یہ درج ذیل مثال سے واضح کیا گیا ہے۔

```
#include<stdio.h>
#include<conio.h>
void fun (int x, int y)
{
    x = 20;
    y = 10;
    printf("Values of x and y in fun(): %d%d", x, y);
}
void main()
{
    int x = 10, y = 20;
    fun (x, y);
    printf("Values of x and y in main(): %d%d", x, y);
    getch();
}
```

#### Output

Values of x and y in fun(): 20 10  
Values of x and y in main ():10 20

سچہو گرام میں فونکشن کی ترتیب کے لیے کس کھدائے کو زہن میں رکھنا چاہیے؟

پروگرام میں فونکشن کی ترتیب کے لیے درج ذیل نکات کو زہن میں رکھنا چاہیے۔

1. اگر کال کیے گئے فونکشن کی ڈیپنیشن کا کال کرنے والے فونکشن کی ڈیپنیشن سے پہلے ظاہر ہوتی ہے تو فونکشن کے سلینچر کی ضرورت نہیں ہے۔
2. اگر کال کیے گئے فونکشن کی ڈیپنیشن کا کال کرنے والے فونکشن کی ڈیپنیشن کے بعد ظاہر ہوتی ہے تو کال کیے گئے فونکشن کا سلینچر اس کال کرنے والے فونکشن سے پہلے لکھنا ضرور ہے۔ مندرجہ ذیل دونوں کوڈز سترکچر زورست ہیں۔

**a)**

```
int add(int, int);
void main()
{
    printf(" %d ",add(4, 5));
}
int add(int a, int b)
{
    return a + b;
}
```

**b)**

```
int add(int a, int b)
{
    return a + b;
}
void main()
{
    printf(" %d ",add(4, 5));
}
```

سچہو ایک فونکشن (prime) کیسی جو ایک نمبر کو ان پٹ کے طور پر لیتا ہے اور اگر ان پٹ نمبر ام ہے تو 0 ریٹن کرے اس فونکشن کو main() میں استعمال کریں۔

### Program:

```
#include <stdio.h>
#include <conio.h>
int prime(int n)
{
    for (int i = 2 ; i < n ; i++)
        if(n % i == 0)
            return0;
            return1;
}
void main ()
{
    clrscr();
    int x;
    printf ("Please enter a number: ");
    scanf ("%d", &x);
    if(prime(x))
        printf("%d is a Prime Number : " ,x);
    else
        printf("%d is not a Prime Number " ,x);
    getch();
}
```

کس: ایک ایسا فنکشن لکھیں جو ثابت نمبر کو ان پت کے طور پر لے اور 0 سے لے کر اس نمبر تک نمبروں کو جمع کرے اور حاصل جمع ریٹرن کرے۔

**Program:**

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
int digitsSum(int n)
{
    int sum = 0;
    for(int i = 0; i <= n; i++)
    {
        sum = sum + i;
    }
    return sum;
}
void main()
{
    clrscr();
    int number;
    printf("Please enter a positive number: ");
    scanf("%d", &number);
    if(number >= 0)
    {
        int sum = digitsSum (number);
        printf ("The sum of numbers upto given number is %d", sum);
    }
    else
        printf("You entered a negative number: ");
    getch();
}
```

## فکشنز

151

س1: کچھ الاتھائی سوالات۔

D	C	B	A	سوال	نمبر
A دنوں C اور A	بیو رڈیفارٹ	سرورڈیفارٹ	ایڈ مونڈیفارٹ	فکشنز بلت ان یا _____ ہو سکتے ہیں۔	1
حکمراری	حکمرار پر جنی	بلت ان	بیو رڈیفارٹ	وہ فکشن جو C میئندرڈ لابرری میں دستیاب ہیں انہیں کہتے ہیں۔	2
آر گیوینش	ارے	ریٹرن ناچس	بازیز	فکشن کو پاس کی گئی قیمتیں کہلاتی ہیں۔	3
آر گیوینش	ارے	ریٹرن ناچس	بازیز	"char cd() {return 'a'}" اس فکشن میں _____ ہے۔	4
تمام	ڈیلگنگ میں آسانی	بار بار استعمال	پڑھے جانے کی صلاحیت	فکشن استعمال کرنے کے فائدے ہیں۔	5
پہلی اور آخری	تمن	وو	ایک	اگر فکشن پاڑی میں تین ریٹرن سٹیٹمنٹس ہوں تو ان میں سے چلے گی۔	6
تمام	ڈیک کرنے	تبدیل کرنے	سمجھنے	پڑھے جانے کی صلاحیت (readability) کوڈ کو کرنے میں مدد دیتی ہے۔	7
اکٹیورڈنگ	ری- رائٹنگ	ڈیفارٹ	کانگ	_____ سے مراد کوڈ ایک اور فکشن میں ٹرانسفر کرنا ہے۔	8

جوابات

1	2	3	4	5	6	7	8
C	B	D	B	D	A	D	A

س2: درج ذیل کی تحریف کریں۔

- 1) فکشنز
  - 2) بلت ان فکشنز
  - 3) فکشنز پیر ایمیزز
  - 4) دوبارہ استعمال
  - 5) فکشن کو کال کرنا
- صفونبر 144 پر ملاحظہ فرمائیں  
صفونبر 144 پر ملاحظہ فرمائیں  
صفونبر 162 پر ملاحظہ فرمائیں  
صفونبر 144 پر ملاحظہ فرمائیں  
صفونبر 147 پر ملاحظہ فرمائیں

س 3: درج ذیل سوالات کے مختصر جوابات لکھیں۔

صخونبر 148 پر ملاحظہ فرمائیں

1) آر گو منس اور بیر امیٹرز میں کیا فرق ہے؟ ایک مثال دیں۔

صخونبر 146 پر ملاحظہ فرمائیں

2) فونکشن ڈیپنیشن کے حصوں کی فہرست لکھیں۔

3) کیا یہ ضروری ہے کہ فونکشن ڈیپنیشن کا ال کی ذیانتاکیں میں ہم آہنگی ہو؟ مثال کے ساتھ جواب کی توثیق کریں۔ صخونبر 148 پر ملاحظہ فرمائیں

4) فونکشنز استعمال کرنے کے فوائد کی وضاحت کریں؟

صخونبر 144 پر ملاحظہ فرمائیں

5) آپ کی ویڈیو یون کے بارے میں کیا جانتے ہیں؟

جواب: ریزن سینٹر فونکشن سے ویڈیو یون کے لیے استعمال ہوتی ہے جیسا فونکشن C لینگوچ میں main فونکشن سے باہر لکھا جاتا ہے۔ یہ فونکشن کے اندر استعمال ہوتا ہے۔ یہ کال کرنے والے فونکشن کو ویڈیو یون کرنے کے لیے استعمال ہوتا ہے۔

س 4: کوڈ کے درج ذیل حصوں میں ایرر زٹلاش کریں۔

a) void sum (int a, int b)  
{  
    return a + b;  
}

**Errors:** If function return type is void  
then its return nothing.

b) void message ()  
{  
    printf ("Hope you are fine :");  
    return 23;

**Errors:** After function definition there is  
no semicolon.

c) int max (int a; int b)  
{  
    if (a > b)  
        return a;  
    return b;  
}

**Errors:** Between parameters use comma  
instead of semicolon.

d) int product (int n1, int n2)  
    return n1\*n2;

**Errors:** Function body curly braces are  
missing.

e) int totalDigits (int x)  
{  
    int count = 0;  
    for (int i = x; i >= 1, i = i/10)  
        count++;  
    return count  
};

**Errors:** At the end of function body  
semicolon is not required.

س 5: کوڈ کے درج ذیل حصوں کی آٹھ پٹ تحریر کریں۔

a) int xyz (int n)  
 {  
     return n + n;  
 }  
 int main()  
 {  
     int p = xyz (5);  
     p = xyz (p);  
     printf("%d ",p);  
 }

**Output:** 10

b) void abc(int a, int b, int c)  
 {  
     int sum = a + b + c;  
 }  
 int main( )  
 {  
     int x = 4, y= 7, z = 23, sum1= 0;  
     abc(x y z);  
     printf ("%d %d %d" x, y,z);  
 }

**Output:** 4 ,7, 23

c) int aa (int x)  
 {  
     int p = x /10;  
     x++;  
     p = p + ( p \* x) ;  
     return p;  
 }  
 int main( )  
 {  
     printf ("We get %d", aa(aa(23)));  
 }

**Output:** 260

d) float f3(int n1, int n2)  
 {  
     n1 = n1 + n2;  
     n2 = n2 - n1;  
     return 0;  
 }  
 int main()  
 {  
     printf ("%f\n", f3(3, 2));  
     printf ("%f\n", f3(10, 6));  
 }

**Output:**  
0  
0

پروگرامنگ کی مشق

مشن نمبر 1: ایک اشیاء کا مرحلہ معلوم کرنے کے لیے int square (int x); فونکشن لکھیں۔

**Program**

```
#include <stdio.h>
#include <conio.h>

int square(int x)
{
    return (x * x);
}

int main()
{
    int num;
    int n;
    printf("Input any number for square : ");
    scanf("%d", &num);
    n = square(num);
    printf("The square of %d is : %d\n", num, n);
    return 0;
}
```

**Output**

```
Input any number for square : 9
The square of 9 is : 81
```

مشن نمبر 2: ایک فونکشن لکھیں جو  $x^y$  معلوم کر کے ریٹن کرے۔ int power(int x,int y);

**Program**

```
#include <stdio.h>
#include <math.h>

int power(int x,int y)
{
    return pow(x,y);
}

int main()
{
    int a,b;
    int n;
    printf("Enter base and power ");
    scanf("%d %d", &a,&b);
    n = power(a,b);
    printf("%d", n);
    return 0;
}
```

**Output**

```
Enter base and power 5
```

4

625

مشن نمبر 3: ایک نمبر کا فکٹریل لٹالے کا نقشن لکھیں۔

**Program**

```
#include <stdio.h>
#include <conio.h>
unsigned int factorial(unsigned int n)
{
    if (n == 0)
        return 1;
    return n * factorial(n - 1);
}
int main()
{
    int num = 5;
    printf("Factorial of %d is %d.", num, factorial(num));
    return 0;
}
```

**Output**

Factorial of 5 is 120.

مشن نمبر 4: ایک نقشن لکھیں جو ایک مثلث کے تین زاویے لے اور بتائے کہ دیے گئے یہ زاویے سچے مثلث کے ہیں یا نہیں۔ ایک سچے مثلث وہ ہے جہاں تین زاویوں کا مجموعہ 180 ہو۔

**Program**

```
#include <stdio.h>
#include <conio.h>
void valid_triangle(int x, int y ,int z)
{
    if (x + y + z == 180)
        printf("Valid Triangle");
    else
        printf("Not valid Triangle");
}
int main()
{
    int a = 45, b = 45, c = 90;
    valid_triangle(a,b,c);
    return 0;
}
```

**Output**

Valid Triangle

مشن نمبر 5: ایک فونکشن کسیں جو رقم اور ائٹرست ریٹ لے اور ائٹرست کی رقم ریٹن کرے۔

### Program

```
#include <stdio.h>
#include <conio.h>
float interest(float x, float y)
{
    float res;
    res= x*(y/100.0);
    return res;
}

int main()
{
    float amount, percentage, result;
    printf("Enter amount and percentage :");
    scanf("%f %f ", &amount, &percentage);
    result = interest(amount,percentage);
    printf("Interest Amount is =%f ",result);
    return 0;
}
```

### Output

Enter amount and percentage : 1000 10 Interest Amount is = 100
--

مشن نمبر 6: کسی نمبر کا ملی پرنت کرنے کے لیے ایک فونکشن کسیں۔

### Program

```
#include <stdio.h>
#include <conio.h>
void table(int num)
{
    for(int i = 1; i<= 10; i++)
        printf("%d x %d = %d\n", num, i, num*i);
}

int main()
{
    int n;
    printf("Enter any number ");
    scanf("%d",&n);
    table(n);
    return 0;
}
```

### Output

Enter any number 6 6 x 1 = 6 6 x 2 = 12 6 x 3 = 18 6 x 4 = 24 6 x 5 = 30 6 x 6 = 36 6 x 7 = 42 6 x 8 = 48 6 x 9 = 54 6 x 10 = 60
--

مشن نمبر 7: ایک فونکشن لکھیں جو ایک نمبر انپٹ لے اور پیسز کے ساتھ اس کے ہر دوسرے پرنسٹ کرے

**Program**

```
#include <stdio.h>
#include <string.h>
void extractDigits(unsigned int num)
{
    int i,len;
    //char pointer declaration
    char temp[5];
    //assig number as string in char pointer
    printf((char*)temp,"%u",num);
    //extract and print each digits
    len= strlen(temp);
    for(i=0; i<len; i++)
        printf("%c ",temp[i]);
    printf("\n");
}
//main program
int main()
{
    unsigned int number=12345;
    //function calling
    extractDigits(number);
    return 0;
}
```

**Output**

1	2	3	4	5
---	---	---	---	---

## فکشنز

158

### کشید الاتجاحی سوالات

نمبر	سوال			
D	C	B	A	
1	کوئی بھی نہیں	ڈیپنیشن پر ابم	مسکے کا عمل	فٹکشن ----- مشکل اور پچیدہ مسائل کو حل کرنے کا عمل کہلاتا ہے۔
2	ایک مسئلہ کا تجزیہ کرنا	ایک مسئلہ کو ٹھان کرنا	ایک مسئلہ کو تقسیم کرنا	تقسیم کرو اور فتح کرو برے مسئلہ کو چھوٹے مسائل میں تقسیم کرنے کے عمل کو۔ کہا جاتا ہے
3	ہیر ایمیز	نکشن	پروگرام	سینٹمنٹس کا ایک بلاگ ہے جو ایک خاص کام برائیم رہتا ہے۔
4	printf	getch	scanf	ایک فٹکشن ہے جو کمپیوٹر اسکرین پر کچھ بھی دکھانے کے لیے استعمال ہوتا ہے۔
5	printf	int	getch	ایک فٹکشن ہے جو یورزور سے ان پت لینے کے لیے استعمال ہوتا ہے۔
6	کوئی بھی نہیں	یوزر ذیفارٹ فٹکشن	اور C دونوں A	C لینگوچ میں فٹکشن کی اقسام _____ ہے۔
7	C فٹکشن کی ٹاپ نہیں ہے	پری- ذیفارٹ فٹکشن	یوزر ذیفارٹ فٹکشن	درج ذیل میں سے کون سا فٹکشن کی قسم نہیں ہے ؟
8	تمام	بلٹ ان فٹکشن	پری- ذیفارٹ فٹکشن	وہ فٹکشن جویں سینڈرڈ لا بسیری میں دستیاب ہیں انہیں کہتے ہیں۔
9	کوئی بھی نہیں	پری- ذیفارٹ فٹکشن	یوزر ذیفارٹ فٹکشن	کون سے فٹکشن ریاضی کے حساب کتاب، سڑنگ آپریشن ان پت / آؤٹ پت آپریشن وغیرہ کے لیے استعمال ہوتے ہیں ؟
10	بلٹ ان فٹکشن	یوزر ذیفارٹ فٹکشن	آر گیومنٹ فٹکشن	کانسٹنٹ فٹکشن scanf اور printf میں۔
11	ریٹرنگ فٹکشن	دونوں فٹکشن	یوزر ذیفارٹ فٹکشن	وہ فٹکشن جو پروگرامر بناتا ہے انہیں _____ کہتے ہیں۔
12	تمام	پڑھنے کی الہیت	کاموں کو الگ کرنا	فٹکشن کے فوائد کیا ہیں ؟
				دوبارہ استعمال

## فناکشنز

159

پڑھنے کی البیت	قابل رسائی	دوبارہ استعمال	مسئلے کی چیزیں کی سے تمثیل	مندرجہ ذیل میں سے کون فناکشن کا فائدہ نہیں ہے؟	13
کوئی نہیں	کاموں کو الگ کرنا	پڑھنے کی البیت	دوبارہ استعمال	فناکشن: کوڈ کو دوبارہ استعمال میں لاستے ہیں۔	14
فناکشن کی عملیجہی	کوڈ کی تبدیلی	کاموں کو الگ کرنا	مسئلہ کی عملیجہی	فناکشن ہمیں ایک فناکشن کے کوڈ کو دوسرے فناکشن کے کوڈ سے الگ کرنے کی اجازت دیتے ہیں۔	15
شیشنس	فناکشن	ویری ایبلز	پروگرام	پروگرام کو چوتھے چھوٹے حصوں میں تقسیم کرتے ہیں اور اس طرح مسئلے کی چیزیں کم ہو جاتی ہے	16
پڑھنے کی البیت	رسائی	دوبارہ استعمال	وجہیہی	پروگرام کو کئی فناکشن میں تقسیم کرنا پر پروگرام کی بہتر بناتا ہے۔	17
سیکھ پڑھ	انڈیکس	ولیو	چیر ایمسز	فناکشن کی ان پٹس کو فناکشن کے کہا جاتا ہے۔	18
آؤٹ پٹ ولیو	ریزن ولیو	ولیوان پٹ ولیو	انڈیکس	فناکشن کی آؤٹ پٹ کو اس کی کہا جاتا ہے۔	19
فناکشن استعمال کرتا ہے	فناکشن ولیو	فناکشن گلینچر	فناکشن انڈیکس	فناکشن کی ان پٹ اور آؤٹ پٹ کی وضاحت کے لیے استعمال ہوتے ہیں۔	20
تم	int largest (int, int, int);	float area (float);	int square (int);	ان میں سے کوئی فناکشن گلینچر کی مثالیں ہیں؟	21
int is Vowel (char);	float area (float);	int largest (int, int, int);	int square (int);	ایک فناکشن جو ایک اشیاء کو بطور ان پٹ لیتا ہے اور اس کا مریع ریزن کرے گا اس کے فناکشن گلینچر ہوتے ہیں۔	22
None	float perimeter (float, float);	int isVowel (char);	float area (float);	ایک فناکشن جو ایک مستطیل کی لمبائی اور پورائی کو بطور ان پٹ لیتا ہے اور مستطیل کا چیر ایمسز ریزن کرے گا اس فناکشن کے گلینچر ہوتے ہیں۔	23
float area (float);	int square (int);	int largest (int, int, int);	int isVowel (char);	ایک فناکشن جو تم ان شیجز کو بطور ان پٹ لیتا ہے اور ان میں سب سے بڑی قیمت ریزن کرے گا اس کے فناکشن گلینچر ہیں۔	24

All	int isVowel (char);	int is Vowel (char);	float area (float);	ایک فنکشن جو دائرے کے رادس کو بطور ان پٹ لیتا ہے اور دائرے کا رقم ریٹن کرے گا اس کے فنکشن سلیچر ہوتے ہیں۔	25
All	int isVowel (char);	float area (float);	char area (float);	ایک فنکشن جو ایک کریکٹر کو بطور ان پٹ لیتا ہے اور اگر حرف علات (vowel) ہے تو 1 ریٹن کرے گا ورنہ 0 ریٹن کرے گا اس کے فنکشن سلیچر ہیں۔	26
کوئی نہیں	فنکشن کے ہیڈر	فنکشن کا سیٹ	باذی کا فنکشن	ان شیٹشنس کا سیٹ ہوتا ہے جنہیں چلا کر فنکشن ایک خاص کام سر انجام دیتا ہے۔	27
4	3	2	1	ایک فنکشن سے زیادہ قیمتیں ریٹن نہیں کر سکتے۔	28
function (value1, value2,... ,value n);	function_ name{value1, value2,, value3};	function_name (value1,value2, ...,valueN);	function_body {value1,value2, ,value n};	درجن ذیل فنکشن کو کال کرنے کا عمومی ذہنچہ ہے۔	29
آر گیومنٹ	سینٹ	فنکشن	فرمیٹر	فنکشن کو دی جانے والی قیمتیں کہتے ہیں۔	30
انڈسکس	پیرامیٹر	سکرپٹ	پر اسکرپٹ	ویری ایبلز جو قیمتیں وصول کرتے ہیں انہیں فنکشن کا کہا جاتا ہے۔	31

٦٤٢

## مختصر سوالات

س 1: مسئلہ کا حل کیا ہے؟

جواب: مشکل اور چیزیں مسئلے کو حل کرنے کے عمل کو مسئلہ حل کرنا کہا جاتا ہے۔ کسی مسئلے کو حل کرنے کے لیے ایک منظم حکمت عملی اپنائی پڑتی ہے۔

س 2: قسم اور فحی وضاحت کریں؟

جواب: قسم کرو اور فحی کی حکمت عملی میں ایک چیزیں مسئلے کو چھوٹے چھوٹے مسائل میں تقسیم کیا جاتا ہے اور ان کا حل نکال کر ایک چیزیں مسئلے کا حل نکال لیا جاتا ہے۔ اس طرح ہمارے لیے ایک وقت میں ایک چھوٹے مسئلے پر توجہ مرکوز کرنا آسان ہو جاتا ہے جبکہ اس کے کہ ہم ہر وقت پورے مسئلے کے بارے میں سوچیں۔

س 3: نقش کی تعریف کریں؟

جواب: نقش شیئننس کا ایک بلاک ہے جو ایک خاص کام انجام دیتا ہے۔ ہر پروگرام میں ایک میں نقش ہوتا ہے جو یہ زر کے کاموں کو سر انجام دیتا ہے۔ اسی طرح ہم دوسرے نقشز کو لے سکتے ہیں اور انہیں کمی بار استعمال کر سکتے ہیں۔

س 4: نقش کی اقسام کے ہم کیسیں؟

جواب: بنیادی طور پر نقش کی دو اقسام ہیں۔

1) بلٹ ان نقشز (Built-in Function)      2) یوزر ڈینیڈ نقشز (User-defined Function)

س 5: بلٹ ان نقشز کی تعریف کریں؟

جواب: جن نقشز کی پہلے سے ہی پروگرام لیٹکنگ بھی میں وضاحت کی جاتی ہے انہیں بلٹ ان نقش کہا جاتا ہے۔ ان نقشز کو لا بھر بری نقش بھی کہا جاتا ہے۔ یہ نقشز لیٹکنگ کے ایک حصے کے طور پر دستیاب ہیں۔ یہ نقشز ریڈی میڈ پروگرام ہیں۔

س 6: یوزر ڈینیڈ نقشز کی تعریف کریں؟

جواب: وہ نقشز جو پروگرام بتاتا ہے انہیں یوزر ڈینیڈ نقشز کہتے ہیں۔ یہ نقشز یوزر کی ضرورت کے مطابق بتائے جاتے ہیں۔

س 7: نقشز کے فوائد کیسیں؟

جواب: نقشز ہمیں درج ذیل فوائد فراہم کرتے ہیں۔

1. دوبارہ استعمال      2. کام کو الگ کرنا      3. پڑھنے کی امیت      4. مسئلے کی چیزیں گی سے نہ نہیں

س 8: دوبارہ استعمال کی وضاحت کریں؟

جواب: نقشز کے ذریعے ہم کو دوبارہ استعمال کر سکتے ہیں۔ اس کا مطلب یہ ہے کہ جب بھی ہمیں ایک نقش کی فکشنیلیٹی (functionality) کی ضرورت ہو تو اس نقش کو کمال کر سکتے ہیں۔ ہمیں بار بار شیئننس کا ایک ہی جمود لکھنے کی ضرورت نہیں ہے۔

س 9: کام کو الگ کرنے سے کیا مراد ہے؟

جواب: نقشز ہمیں ایک کام کے کوڈ کو دوسرے کاموں کے کوڈ سے الگ کرنے کی اجازت دیتے ہیں۔ اگر ہمیں ایک نقش میں کوئی مسئلہ دریافت ہے تو ہم ہمیں اس مسئلے کو دور کرنے کے لیے پورے پروگرام کو چیک کرنے کی ضرورت نہیں ہے۔ ہمیں صرف ایک نقش پر توجہ دینے کی ضرورت ہے۔

س10: پڑھنے کی الہیت کی وضاحت کریں؟

جواب: پروگرام کو کمی فناشر میں تقییم کرنے پر دگر ایام کی پڑھنے کی الہیت کو بہتر بنتا ہے۔

س11: مسئلہ کی جیجیدگی سے مٹتا کی اصطلاح کی وضاحت کریں؟

جواب: اگر ہم پورے پروگرام کو ایک پروپریتی کے طور پر لکھیں تو پروگرام کی دیکھ بھال کرنا ممکن ہو جاتا ہے۔ فناشر پروگرام کو چھوٹے چھوٹے حصوں میں تقییم کرتے ہیں اور اس طرح مسئلے کی جیجیدگی کم ہو جاتی ہے۔

س12: فناشر گنجائی کا استعمال کیا ہے؟

جواب: فناشر کے سینچر (signature) فناشر کی ان پٹ اور آٹھ پٹ کی وضاحت کے لیے استعمال ہوتے ہیں۔

س13: فناشر کا ہر ایمیڈ کا ہے؟

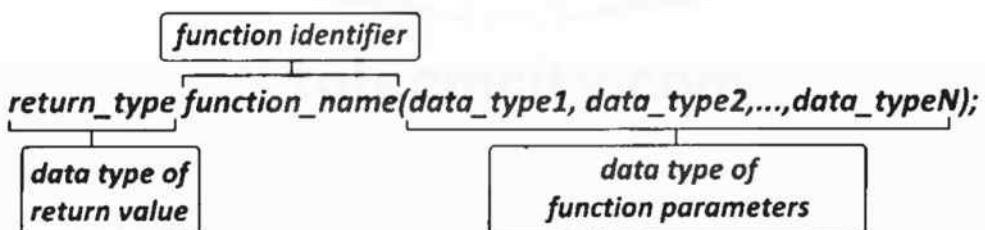
جواب: فناشر شیئننس کا ایک بلاک ہے جو کچھ ان پٹ حاصل کرتا ہے اور کچھ آٹھ پٹ فراہم کرتا ہے۔ فناشر کی ان پٹوں کو فناشر کے ہر ایمیڈ کہا جاتا ہے۔

س14: فناشر کی ریزن ولیو سے آپ کا کام مطلب ہے؟

جواب: فناشر شیئننس کا ایک بلاک ہے جو کچھ ان پٹ حاصل کرتا ہے اور کچھ آٹھ پٹ فراہم کرتا ہے۔ فناشر کی آٹھ پٹ کو اس کی ریزن ولیو کہا جاتا ہے۔ ایک فناشر میں ایک سے زیادہ ہر ایمیڈ زہو کتے ہیں لیکن یہ ایک سے زیادہ قیمت ریزن نہیں کر سکتے۔

س15: فناشر کے گنجائی کا عمومی دھانچہ لکھیں۔

جواب:



س16: int square(int); فناشر کیا کام کرتا ہے؟

جواب: ایک فناشر جو ایک انسٹیجیر کو بطور ان پٹ لیتا ہے اور اس کا مردی ریزن کرے گا۔

س17: int float perimeter (float, float); فناشر کیا کام کرتا ہے؟

جواب: ایک فناشر جو مستطیل کی لمبائی اور چوڑائی کو بطور ان پٹ لیتا ہے اور مستطیل کا احاطہ ریزن کرے گا۔

س18: int largest (int, int, int); فناشر کیا کام کرتا ہے؟

جواب: ایک فناشر جو تین انسٹیجیر کو بطور ان پٹ لیتا ہے اور ان میں سب سے بڑی قیمت ریزن کرے گا۔

س19: float area (float); فناشر کیا کام کرتا ہے؟

جواب: ایک فناشر جو دائیے کے رداں کو بطور ان پٹ لیتا ہے اور دائیے کا قبر ریزن کرے گا۔

س 20: int isVowel (char); فونکشن کیا کام کرتا ہے؟

جواب: ایک فونکشن جو ایک کریکٹر کو بطور ان پٹ لیتا ہے اور اگر حرف علات (vowel) ہے تو 1 ریٹن کرے گا اور نہ 0 ریٹن کرے گا۔

س 21: فونکشن کوڈ یہاں کرنے کا عمومی ڈھانچہ بیان کریں؟

جواب: ایک فونکشن کے سلیکنپر سے یہ نہیں پڑھ پڑھا کر وہ کس کام کے لیے ہے۔ اس کے لیے فونکشن ڈیفینیشن ہوتی ہے۔ ایک فونکشن ڈیفینیشن کا ڈھانچہ کچھ اس طرح سے ہے۔

```
return_type function_name (data_type var1, data_type var2, ..., data_type varN)
{
    Body of the function
}
```

س 22: فونکشن ڈیفینیشن کی وضاحت کے لیے مثال لکھیں۔

جواب: درج ذیل میل ایک فونکشن (showPangram()) کی وضاحت کرتی ہے جو کوئی ان پٹ نہیں لیتا اور کچھ بھی ریٹن نہیں کرتا اس کپیوٹر سکرین پر "A Quick Brown Fox Jumps over the Lazy Dog" دکھاتا ہے۔

<pre>void showPangram() {     printf("\n A quick brown fox jumps over the lazy dog.\n"); }</pre>	function name
--	---------------

جیسا کہ مذکورہ فونکشن کچھ ریٹن نہیں کرتا اس طرح فونکشن کی ریٹن ناپ void ہے۔

س 23: کیا فونکشن ایک سے زیادہ ٹائیپس و اپس کر سکتا ہے؟

جواب: نہیں ایک فونکشن ایک سے زیادہ ٹائیپس و اپس نہیں کر سکتا۔

س 24: فونکشن کا ال کرنے کے لیے عمومی ڈھانچہ بیان کریں؟

جواب: ہمیں کسی فونکشن کو کا ال کرنے کی ضرورت ہے تاکہ وہ پروگرام کو سونپا گیا کام انجام دے۔ مندرجہ ذیل عمومی ڈھانچہ ہے جو فونکشن کا ال کرنے کے لیے استعمال ہوتا ہے۔

**function\_name(value1, value2, ..., valueN);**

س 25: آر گیو منش کی وضاحت کریں؟

جواب: فونکشن کو دی جانے والی ٹائیپس آر گیو منش کہلاتی ہیں۔

س 26: فونکشن کے آر گیو منش اور پیر ایمیٹر میں فرق کریں۔

جواب: جو ٹائیپس فونکشن کو پاس کی جاتی ہیں وہ آر گیو منش کہلاتی ہے جبکہ فونکشن ڈیفینیشن میں جن دری ایبلز میں یہ ٹائیپس جاتی ہیں انہیں فونکشن کے پیر ایمیٹر کہا جاتا ہے۔

س 27: پروگرام میں فکشنز کی ترتیب کے لیے کس تکمیل کو ذہن میں رکھنا چاہیے؟

جواب: پروگرام میں فکشنز کی ترتیب کے لیے درج ذیل نکات کو ذہن میں رکھنا چاہیے۔

1. اگر کال کے گئے فکشن کی ذیپیشیشن کا ال کرنے والے فکشن کی ذیپیشیشن سے پہلے ظاہر ہوتی ہے تو فکشن کے سلیکنپر کی ضرورت نہیں ہے۔

2. اگر کال کے گئے فکشن کی ذیپیشیشن کا ال کرنے والے فکشن کی ذیپیشیشن کے بعد ظاہر ہوتی ہے تو کال کے گئے فکشن کا سلیکنپر اس کا ال کرنے والے فکشن سے پہلے لکھنا ضروری ہے۔

س 28: فکشن میں کتنے آر گیو منش استعمال کے جائزے ہیں؟

جواب: سی لینگوچ میں آر گیو منش کی تعداد پر کوئی پابندی نہیں ہے لیکن 8 سے زیادہ آر گیو منش کو ترجیح نہیں دی جاتی۔

س 29: مدرج ذیل پروگرام کی آٹھ پٹ کیا ہے؟

```
main()
{
int x=20;
int y=10;
swap(x,y);
printf("%d %d",y,x+2);
}
swap(int x, int y)
{
int temp;
temp =x;
x=y;
y=temp;
}
```

**Output:**

10 22 (duplicate copy of arguments are generated on calling a function)

س 30: درج ذیل کوڈ کی آٹھ پٹ کیا ہوگی؟

```
output()
{
printf("%p",output);
}
```

جواب: کچھ ایڈریس پر نٹ کے جائیں گے کیونکہ فکشن کے نام صرف ایڈریس ہے۔ اسی طرح آٹھ پٹ ( ) بھی ایک فکشن ہے اور اس کا ایڈریس پر نٹ کیا جائے گا۔

س31: درج زمیں کوڈ کی آٹھ پٹ کیا ہوگی؟

```
main()
{
int i;
printf("%d",scanf("%d",&i)); // value 10 is given as input here
}
```

جواب: کامیابی سے پڑھی گئی اشیاء کی تعداد بطور scanf میں 1 ہے۔ تو پڑھی گئی اشیاء کی تعداد 1 ہے۔

س32: ہی پر د گرام کی آٹھ پٹ کیا ہوگی؟

```
#include<stdio.h>
int main()
{
function();
return 0;
}
void function()
{
printf("Function in C is awesome");
}
Output: Compilation error.
```

س34: ہی پر د گرام میں ایور کیا ہوگا؟

س33: ہی پر د گرام میں ایور کیا ہوگا؟

```
int max (int a; int b)
{
    if (a > b)
        return a;
    return b;
}
```

ایور: جہاں ایمپریز کے درمیان سبھی کالن کی بجائے کوئی نہ آئے گا۔

```
void message ();
{
printf ("Hope you are fine : ");
return 23;
```

ایور: فکشن ذیچیشن کے بعد سبھی کالن نہیں آئے گا۔