باب 2

یوزر انٹریکشن

تدریسی مقاصد:(Students Learning Outcomes)

- آؤٹ پٹ فنکشن جیسے: ()Printf کا استعمال
- ان پُٹ فنکشن جیسے:
  - scanf( )
  - ()getch کا استعمال
- سٹیٹمنٹ ٹرمینیٹر (سیمی کولن) کا استعمال
- فارمنٹ سپیسفائرز کی تعریف
  - انٹیجر (%i)
  - ڈیسیمل (%d)
  - فلوٹ (%f)
  - کریکٹر (char) (%c)
- اسکیپ سیکوئنس کی تعریف
- درج ذیل اسکیپ سیکوئنسز کی پروگرامنگ کی مثالوں کے ساتھ وضاحت:
  - نیولائن (/n)
  - ٹیب (/t)
- ارتھمیٹک اوپریٹر کی تعریف
- درج ذیل ارتھمیٹک اوپریٹرز کا استعمال
  - جمع (+)
  - تفریق (-)
  - ضرب (*)
  - تقسیم (/)
  - ریمنڈر/باقی (%)

باب 2

- اسائنمنٹ اوپریٹرز کا استعمال
- ریشنل اوپریٹرز کی تعریف
- درج ذیل ریشنل اوپریٹرز کا استعمال
  - سے کم (<)
  - سے زیادہ (>)
  - سے کم یا برابر (<=)
  - سے زیادہ یا برابر (>=)
  - کے برابر (=)
  - کے برابر نہیں (!=)
- منطقی اوپریٹر کی تعریف
- درج ذیل منطقی اوپریٹر کا استعمال
  - AND (&&)
  - OR (||)
  - NOT (!)
- اسائنمنٹ اوپریٹر (=) اور برابر کے اوپریٹر (= =) میں فرق
- یونری اور بائنری اوپریٹرز میں فرق
- اوپریٹرز کی ترجیح کی ترتیب کی تعریف اور وضاحت

تعارف (Introduction)

کمپیوٹر ایک ایسا آلہ ہے جو ڈیٹا کو بطور ان پٹ لیتا ہے اس ڈیٹا کو پروسیس کرتا ہے اور آؤٹ پٹ دیتا ہے۔ اسی لیے تمام پروگرامنگ کی زبانوں (Programming Language) میں ڈیٹا کی ان پٹ، آؤٹ پٹ اور پروسیسنگ کے لیے ہدایات ضروری جاتی ہیں ۔ اس باب میں ہم C-لینگوئج (C-Language) کے مختلف پری بلٹ (Pre-Built) ان پٹ، آؤٹ پٹ فنکشنز (Funcitons) کے بارے میں بھی پڑھیں گے۔ ہم ڈیٹا کو پروسیس (Process) کرنے کے لیے استعمال ہونے والے مختلف اوپریٹرز (Operators) کے بارے میں پڑھیں گے۔

1.2 اِن پٹ/آؤٹ پٹ (I/O) فنکشن (Input/Output Functions):

ہمیں پروگرام لکھتے ہوئے ان پٹ دینے اور آؤٹ دکھانے کے لیے ایک لائحہ عمل درکار ہوتا ہے۔ ان پٹ/آؤٹ پٹ آپریشنز کے لیے ہر پروگرامنگ لینگوئج کے اپنے کی ۔ورڈز (Keywords) یا سٹینڈرڈ لائبریری فنکشنز Standard Library) (Functions ہوتے ہیں۔ C-لینگوئج (C-Language) آؤٹ پٹ دکھانے کے لیے printf فنکشنز اور صارف (User) سے ان پٹ لینے کے لیے scanf فنکشن مہیا کرتی ہے۔ اس سیکشن (Section) میں ہم یہ دو فنکشن پڑھیں گے۔

Printf() 1.1.2:

Printf C- پروگرامنگ لینگوئج میں سکرین پر آؤٹ پٹ دکھانے کا ایک بلٹ اِن (Built-in) فنکشن ہے. اس کا نام پرنٹ فارمیٹڈ (Print Formatted) سے نکلا ہے کیوں کہ یہ سکرین پر فارمیٹ کی ہوئی آؤٹ پٹ دکھانے کے لیے استعمال ہوتا ہے۔ پچھلے باب میں ڈیٹا کی جن اقسام کا ذکر کیا گیا ہے وہ سب printf فنکشن سے ڈسپلے کی جا سکتی ہیں مثال کے طور پر نیچے دیے گئے پروگرام کو دیکھیں۔

EXAMPLE CODE 2.1

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

اس مثال میں سکرین پر Hello World دکھانے کے لیے printf فنکشن استعمال کیا گیا ہے۔ ہم واوین (" ") کے درمیان میں جو بھی لکھتے ہیں وہ سکرین پر آ جاتا ہے۔

سرگرمی 2.1:

دیے گئے کوڈ کی آؤٹ پٹ لکھیں۔

```
# include<stdio.h>
void main()
 {
       printf(“I am UPPERCASE and this is lowercase”);
 }
```

سرگرمی 2.2:

سکرین پر اپنے نام کا پہلا حصّہ اپر کیس (Upper Case) اور دوسرا حصّہ لوئر کیس (Lower Case) میں دکھانے کے لیے پروگرام لکھیں۔

2.1.2 فارمیٹ سپیسفائر (Fromate Specifier)

اگر ہم ایک ویری ایبل کی قیمت دکھانا چاہیں تو آیئے! ہم ایک ویری ایبل ڈکلیر کرتے ہیں اور پھر دیکھتے ہیں printf کیسے کام کرتا ہے۔

```
int age = 35;
```

اب ہم اس ویری ایبل کی قیمت سکرین پر دکھانا چاہتے ہوں تو ہم سٹیٹمنٹ (Statement) لکھیں گے۔

```
printf("age");
```

لیکن اس سے ہمارا مقصد حاصل نہیں ہوا کیونکہ یہ سکرین پر دکھاتا ہے۔

```
age
```

یہ ویری ایبل age میں محفوظ کی ہوئی قیمت نہیں دکھاتا بلکہ جو بھی printf کی واوین میں لکھا ہے وہ ہی دکھاتا ہے۔ درحقیقت ہمیں فارمیٹ سپیسفائر کو استعمال کرتے ہوئے اس ڈیٹا کے فارمیٹ کی وضاحت کرنی پڑے گی جو ہم سکرین پر دکھانا چاہتے ہیں۔ C- لینگویج میں ڈیٹا کی مختلف اقسام کے لیے جو فارمیٹ سپیسفائر ہیں وہ ٹیبل 2.1 میں دیے گئے ہیں۔

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

Table 2.1: ان پٹ/آؤٹ پٹ آپریشنز کے لئے فارمیٹ سپیسفائر

فرض کریں ہم int ٹائپ کا ڈیٹا دکھانا چاہتے ہیں تو ہمیں فارمیٹ سپیسفائر d% یا i% کو استعمال کرتے ہوئے printf میں اس کی وضاحت کرنی ہوگی۔ اسی طرح سے float ٹائپ کے لئے ہم f% استعمال کریں گے۔ یہ درج ذیل مثال میں واضح کیا گیا ہے۔

EXAMPLE CODE 2.2

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

Output:

```
My age is 35 and my height is 5.800000
```

اوپر دی گئی مثال میں ہم دیکھ سکتے ہیں کہ آؤٹ پٹ دکھاتے ہوئے پہلے فارمیٹ سپیسفائر کی جگہ واوین کے بعد دیے گئے ڈیٹا میں سے پہلے ویری ایبل یعنی age کی قیمت آ گئی ہے اور دوسرے فارمیٹ سپیسفائر کی جگہ دوسرا ویری ایبل آ گیا ہے۔

اہم معلومات:

جب ہم float کی قیمت دکھانے کے لیے f% استعمال کرتے ہیں تو یہ ڈیسمل پوائنٹ (Decimal Point) کے بعد چھ ہندسے دکھاتا ہے۔ اگر ہم چاہتے ہیں کہ ڈیسمل پوائنٹ کے بعد آنے والے ہندسوں کی تعداد خود بتائیں تو ہم nf% لکھ سکتے ہیں جس میں n سے مراد ہندسوں کی تعداد ہے۔ اوپر دی گئی مثال میں اگر ہم یہ سٹیٹمنٹ لکھیں۔

```
printf(“My age is %d and my height is % 2f”, age, height);
```

تو یہ آؤٹ پٹ ہوگی۔

```
My age is 35 and my height is 5.80
```

اہم نوٹ:

فارمیٹ سپیسفائر صرف ویری ایبل کے لیے استعمال نہیں کیے جاتے ہیں دراصل وہ ہر اس ایکسپریشن (Expression) کا نتیجہ دکھانے کے لئے استعمال ہوتے ہیں جس میں ویری ایبل ہوں، کانسٹینٹس ہوں یا دونوں ہوں۔ جیسا کہ اس مثال سے واضح کیا گیا ہے۔

EXAMPLE CODE 2.3

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

Output

```
Sum of 23 and 45 is 68
```

2.1.3 Scanf()

scanf C-لینگویج میں ایک بلٹ اِن فنکشن (Built-in Function) ہے جو صارف سے ویری ایبلز میں ان پٹ لینے کے لیے استعمال ہوتا ہے۔ scanf فنکشن میں ہم فارمیٹ سپیسفائر کی مدد سے ان پٹ ڈیٹا کی متوقع قسم کی وضاحت کرتے ہیں۔ اگر صارف int ٹائپ کا ڈیٹا دیتا ہے تو scanf میں لکھا گیا فارمیٹ سپیسفائر d% یا i% ہونا چاہیے۔ درج ذیل مثال پر غور کریں۔

EXAMPLE CODE 2.4

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

اس مثال میں فارمیٹ سپیسفائر c% کریکٹر ٹائپ کے ویری ایبل کے لیے استعمال کیا گیا ہے۔ صارف کی دی گئی ان پٹ ویری ایبل Grade میں محفوظ کی گئی ہے۔ scanf فنکشن کے دو اہم حصے ہیں جیسا کہ اوپر دیے گئے کوڈ میں دیکھا جا سکتا ہے۔ پہلے حصے میں واوین کے اندر فارمیٹ سپیسفائر کی ایک فہرست ہوتی ہے اور دوسرے حصے میں ویری ایبلز کی فہرست ہوتی ہے۔ جس کے بائیں طرف & لکھا جاتا ہے۔

EXAMPLE CODE 2.5

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

Output:

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

اہم نوٹ:

ہم ایک scanf فنکشن کو استعمال کرتے ہوئے ایک سے زیادہ ان پٹس لے سکتے ہیں۔مثال کے طور پر اس سٹیٹمنٹ کو دیکھیں۔

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

یہ سٹیٹمنٹ دو انٹیجر (integer) ٹائپ کے ویری ایبلز a اور b اور ایک float ٹائپ کے ویری ایبل میں ان پٹ لیتی ہے۔ ہر ان پٹ کے بعد صارف کو space یا enter دبانا ہوتا ہے اور تمام ان پٹس دینے کے بعد enter دبانا لازمی ہوتا ہے۔

سرگرمی 2.3

ایک پروگرام لکھیں جو صارف سے رول نمبر، نمبروں کی پرسنٹیج اور گریڈ بطور ان پٹ لے اور اس طرح سے فارمیٹ کی ہوئی آؤٹ پٹ دکھائے۔

```
Roll No     :        input value
Percentage  :        input value %
Grade       :        input value
```

اہم نوٹ:

scanf فنکشن میں & اوپریٹر کو بھول جانا بہت عام غلطی ہے۔ & اوپریٹر کے بغیر پروگرام چل تو جاتا ہے پر توقع کے مطابق نہیں چلتا۔

getch() :2.1.4

getch() فنکشن صارف سے ایک کریکٹر لینے کے لیے استعمال ہوتا ہے۔ صارف جو کریکٹر ان پُٹ کرتا ہے وہ سکرین پر نہیں ظاہر ہوتا۔ یہ فنکشن عام طور پر پروگرام کی ایگزیکیوشن (execution) کو روکنے کے لیے استعمال کیا جاتا ہے کیونکہ پروگرام اس وقت تک آگے نہیں چلتا جب تک صارف کوئی بٹن نہیں دباتا۔ اس فنکشن کو استعمال کرنے کے لیے ہم پروگرام کے ہیڈر سیکشن (header section) میں ایک لائبریری conio.h شامل کرتے ہیں۔

EXAMPLE CODE 2.6

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

اوپر دی گئی مثال میں پروگرام صارف سے کریکٹر اِن پُٹ کرنے کا مطالبہ کرتا ہے اور پھر پروگرام ایگزیکیوشن ختم کرنے سے پہلے ان پٹ کا انتظار کرتا ہے۔

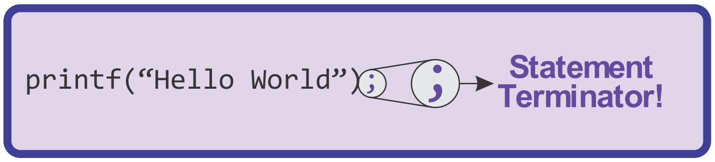
EXAMPLE CODE 2.7

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

اگر ہم یہ پروگرام ایگزیکیوٹ کریں تو ہمیں scanf اور getch فنکشن سے کریکٹر لینے میں فرق کا پتا چلے گا۔ جب ہم scanf کے ذریعے کریکٹر اِن پُٹ کرتے ہیں تو ہمیں پروگرام کو آگے چلانے کے لئے enter دبانا پڑتا ہے۔ لیکن getch میں پروگرام enter دبانے کا انتظار نہیں کرتا۔ یہ فنکشن کریکٹر لیتے ہی اگلی سٹیٹمنٹ ایگزیکیوٹ کرنا شروع کر دیتا ہے۔

2.1.5 سٹیٹمنٹ ٹرمینیٹر (Statement Terminator)

سٹیٹمنٹ ٹرمینیٹر کمپائلر کو یہ بتاتا ہے کہ لائن ختم ہو گئی ہے C-لینگوئج میں سیمی کولن (;) بطور سٹیٹمنٹ ٹرمینیٹر استعمال ہوتا ہے۔
اگر ہم ہر سٹیٹمنٹ کے آخر میں ";" نہ لگائیں تو نتیجتاً error آ جاتا ہے۔



2.1.6 اسکیپ سیکوئنس (Escape Sequence)

مقصد:

اسکیپ سیکوئنس printf فنکشن میں واوین کے درمیان استعمال ہوتے ہیں ان کے ذریعے printf فنکشن معمول کے طریقۂ کار سے ہٹ کر آؤٹ پٹ دکھاتا ہے اس کو سمجھنے کے لئے ہم نیچے دی گئی سٹیٹمنٹ کو دیکھتے ہیں۔

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

اوپر دی گئی سٹیٹمنٹ کی آؤٹ پٹ یہ ہے۔

```
My name is "Ali"
```

اوپر دی گئی مثال میں "\" ایک اسکیپ سیکوئنس ہے یہی وجہ ہے کہ printf "" کو سکرین پر دکھاتا ہے۔

اسکیپ سیکوئنس کی بناوٹ:

اسکیپ سیکوئنس میں دو کریکٹر ہوتے ہیں پہلا کریکٹر ہمیشہ بیک سلیش (back slash) "\" ہوتا ہے اور دوسرا کریکٹر مطلوبہ فنکشنلیٹی کے مطابق آتا ہے۔ بیک سلیش (\) کو اسکیپ کریکٹر (escape character) کہا جاتا ہے کیونکہ یہ ہر اسکیپ سیکوئنس کے ساتھ اسکیپ کی نشان ہی کرنے کے لیے منسلک ہوتا ہے۔ اسکیپ کریکٹر اور اس کے بعد آنے والا کریکٹر سکرین پر نہیں لکھے جاتے لیکن یہ وہ مخصوص کام کرتے ہیں جو ان سے منسوب کیا گیا ہے۔

کیا آپ جانتے تھے؟

اس سیکشن میں جن اسکیپ سیکوئنسز کا ذکر کیا گیا ان کے علاوہ C-لینگوئج کے درج ذیل اسکیپ سیکوئنس بھی عموماً استعمال ہوتے ہیں۔

سیکوئنس	مقصد	سیکوئنس	مقصد
\'	سنگل کوٹ سکرین پر دکھانا (')	\a	الرٹ کی آواز پیدا کرتا ہے
\\	بیک سلیش سکرین پر دکھانا (\)	\b	پچھلا کریکٹر مٹاتا ہے

نیولائین "\n"(New Line)

اسکیپ کریکٹر کے بعد n واضح کرتا ہے کہ کرسر (Cursor) کو اگلی لائن کے شروع پر لے کر جانا ہے۔ یہ اسکیپ سیکوئنس آؤٹ پٹ کو ایک سے زیادہ لائنوں میں پرنٹ کرنے کے لیے استعمال ہوتا ہے۔ اسکیپ سیکوئنس کو مزید سمجھنے کے لیے اس مثال پر غور کریں۔

EXAMPLE CODE 2.8

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

Output

```
My name is Ali.
I live in Lahore.
```

اہم نوٹ:

اسکیپ سیکوئنس کے بغیر اگر ہمارے پاس ایک سے زیادہ printf سٹیٹمنٹس بھی ہوں تو آؤٹ پٹ ایک ہی لائن پر آئے گی۔ اس مثال سے یہ بات واضح ہوتی ہے۔

EXAMPLE CODE 2.9

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

Output

```
My name is Ahmad
```

ٹیب "\t"(Tab)

اسکیپ سیکوئنس t\ ,I/O فنکشن کو بتاتا ہے کہ افقی طور پر اگلے ٹیب سٹاپ (Tab Stop) پر جانا ہے۔ ایک ٹیب سٹاپ آٹھ سپیسز کا مجموعہ ہوتا ہے t\ استعمال کرنے سے کرسر اگلے ٹیب سٹاپ پر چلا جاتا ہے۔

EXAMPLE CODE 2.10

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

Output

```
Name:   Ali
Fname:  Hammad
Marks:  1000
```

2.2 اوپریٹرز (Operators)

کمپیوٹر کے نام سے واضح ہوتا ہے کہ اس کا سب سے اہم کام کمپیوٹیشن (Computation) کرنا ہے۔ ہم پروگرامنگ کے ذریعے ڈیٹا پر کمپیوٹیشنز کرتے ہیں۔ ڈیٹا پر حساب کتاب کرنے کے لیے ہمارے پاس بہت سے ریاضی کے فنکشنز ہیں۔ ہم اپنے پروگراموں میں ریاضی کے آپریشنز بھی استعمال کر سکتے ہیں۔ C- لینگوئج میں ڈیٹا کو ترتیب دینے اور پروسس کرنے کے لیے بہت سے آپریٹرز ہیں۔ کچھ بنیادی آپریٹرز کی فہرست درج ذیل ہیں:

- اسائنمنٹ آپریٹرز (Assignment Operators)
- ریلیشنل آپریٹرز (Relational Operators)
- ارتھمیٹک آپریٹرز (Arithmatic Operators)
- منطقی آپریٹرز (Logical Operators)

1.2.2 اسائنمنٹ آپریٹرز (Assignment Operators)

اسائنمنٹ آپریٹر ویری ایبل کو قیمت تفویض (Assign) کرنے کے لیے یا ایک ویری ایبل کی قیمت دوسرے ویری ایبل میں رکھنے کے لیے استعمال ہوتا ہے۔ C- لینگوئج میں برابری کی علامت (=) بطور اسائنمنٹ آپریٹرز استعمال ہوتا ہے۔
C- لینگوئج میں "=" کا سمبل اسائنمنٹ اوپریٹر کے لیے استعمال کیا جاتا ہے۔ مثال کے طور پر:

```
int sum=5;
```

کوڈ کی یہ لائن ایگزیکیوٹ ہونے کے بعد ویری ایبل sum میں قیمت 5 محفوظ ہوجائے گی۔ اب ایک مثال دیکھتے ہیں۔

int sum=6;

int var=sum;

پہلے ویری ایبل sum کی قیمت 6 رکھی پھر اگلی لائن میں sum کی قیمت ویری ایبل var سے منسوب کر دی۔

2.1 پروگرامنگ ٹائم (Programming Time)

ایک پروگرام لکھیں جو دو انٹیجر متغیرات کی قیمتوں کا آپس میں تبادلہ کر دے۔

```
void main()
{
        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);
}
```

2.2.2 ارتھمیٹک آپریٹرز (Arithmatic Operators)

ارتھمیٹک آپریٹرز ڈیٹا پر ارتھمیٹک آپریشنز انجام دینے کے لیے استعمال ہوتے ہیں۔ ٹیبل 2.2 میں ارتھمیٹک آپریٹرز اور ان کی تفصیل دی گئی ہے:

اوپریٹرز	نام	تفصیل
/	تقسیم کا آپریٹر	یہ بائیں طرف کی قیمت کو دائیں طرف والی پر تقسیم کرتا ہے۔
*	ضرب کا آپریٹر	یہ دو قیمتوں کو ضرب کرتا ہے
+	جمع کا آپریٹر	یہ دو قیمتوں کو جمع کرتا ہے
-	تفریق کا آپریٹر	یہ بائیں طرف کی قیمت سے دائیں طرف والی قیمت تفریق کرتا ہے
%	ماڈولس کا آپریٹر	یہ بائیں طرف کی قیمت کو دائیں طرف کی قیمت پر عددی تقسیم کرنے کے بعد بچنے والی رقم دیتا ہے۔

ٹیبل 2.2 میں ارتھمیٹک آپریٹرز

تقسیم (Division):

تقسیم کا اوپریٹر بائیں اوپرینڈ (Operand) کی قیمت کو دائیں آوپرینڈ کی قیمت پر تقسیم کرتا ہے۔ مثال کے طور پر اس سٹیٹمنٹ کو دیکھیں۔

اس سٹیٹمنٹ کے بعد ویری ایبل result کی قیمت 1.5 ہے۔

```
float result=3.0/2.0;
```

اہم نوٹ:

اگر دونوں اوپرینڈ انٹیجر (int) ٹائپ کے ہوں گے تو تقسیم کا جواب بھی int ٹائپ کا ہوگا۔ جواب انٹیجر دینے کے لیے "باقی" کو نظرانداز کر دیا جاتا ہے کوڈ کی اس لائن کو دیکھیں۔

```
float result = 3.0 / 2;
```

اگر دونوں قیمتیں int ٹائپ کی ہوں گی تو جواب بھی انٹیجر ہوگا یعنی 1-جب یہ قیمت (1) فلوٹ ٹائپ کے ویری ایبل result سے منسوب کی جائے تو یہ (1) فلوٹ ٹائپ میں تبدیل ہوجاتا ہے۔اس لیے ویری ایبل result میں 1.0 قیمت محفوظ ہوتی ہے۔
اگر ہم چاہتے ہیں کے جواب درست اور جامع ہو تو ضروری ہے کہ کوئی اوپرینڈ فلوٹ ٹائپ کا ہی ہو۔کوڈ کی اس لائن کو دیکھیں۔

```
float result = 3.0 / 2;
```

اوپر دی گئی مثال میں متغیر result میں قیمت 1.5 محفوظ ہے۔

2.2 پروگرامنگ ٹائم (Programming Time)

```
/*This program takes as input the price of a box of chocolates and
the total number of chocolates in the box. The program finds and
displays the price of one chocolate.*/
# include <stdio.h>
void main ()
{
      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 chocolate is %f", unit_price);
}
```

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 chocolate is 3.000000
```

ضرب (Multiplication):

ضرب کا آپریٹر (*) ایک بائنری آپریٹر ہے۔ جو دو اعداد کو ضرب کرتا ہے درج ذیل سٹیٹمنٹ دیکھیں:

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

اس سٹیٹمنٹ کے چلنے کے بعد ویری ایبل multiply کی قیمت 25 ہے۔

2.3 پروگرامنگ ٹائم (Programming Time)

```
/* Following program takes as input the length and width of a rectangle. Program calculates and displays the area of rectangle on screen. */
# include<stdio.h>
void main ()
{
        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);
}
```

Output

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

سرگرمی: 2.4

ایک پروگرام لکھیں جو چوکور کے ایک طرف کی لمبائی ان پٹ لے اور اس کے رقبے کا حساب لگائے۔

جمع (Addition):

جمع کا آپریٹر (+) دو آپرینڈز کو جمع کرتا ہے۔ اس سٹیٹمنٹ کو دیکھیں:

int add=10+10;

نتیجتاً ویری ایبل add کی قیمت 20 ہے۔

2.4 پروگرامنگ ٹائم (Programming Time)

```
/* This program takes marks of two subjects from user and displays
the sum of marks on console. */
#include <stdio.h>
void main ()
{
        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);
}
```

Output

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

سرگرمی : 2.5

ایک پروگرام لکھیں جو جار A اور B میں موجود گیندوں کی تعداد ان پٹ لے اور گیندوں کی کل تعداد سکرین پر دکھائے۔

اہم معلومات:

سٹیٹمنٹ ;a=a+1 ویری ایبل a کی قیمت میں 1 کا اضافہ کرنے کے لیے استعمال ہوتی ہے۔ C- لینگوئج میں اس سٹیٹمنٹ کو ;++a یا ;a++ بھی لکھ سکتے ہیں اسی طرح ;--a یا ;a-- ویری ایبل a کی قیمت میں 1 کی کمی کرنے کے لیے استعمال ہوتا ہے۔

تفریق (Subtraction):

تفریق کا آپریٹر بائیں آپرینڈ میں سے دائیں آپرینڈ کو تفریق کرتا ہے آیئے! درج ذیل سٹیٹمنٹ کو دیکھتے ہیں۔

```
int result=20-15;
```

تفریق کے بعد ویری ایبل result کی قیمت 5 ہے۔

سرگرمی 2.6:

ایک پروگرام لکھیں جو قمیص کی اصل قیمت اور ڈسکاؤنٹ پرسنٹیج صارف سے ان پٹ لے اور قمیص کی اصل قیمت، قیمت پہ کیا گیا ڈسکاؤنٹ اور ڈسکاؤنٹ کے بعد والی قیمت سکرین پر دکھائے۔

ماڈولس اوپریٹر (Modulus Operator):

ماڈولس آپریٹر (%) بائیں آپرینڈ کو دائیں آپرینڈ پر تقسیم کرتا ہے اور تقسیم کے بعد بچنے والی "باقی" رقم لوٹاتا ہے۔ ماڈولس اوپریٹر انٹیجر ڈیٹا ٹائپس پر کام کرتا ہے۔

```
int remaining = 14 % 3;
```

اگر ہم 14 کو 3 پر تقسیم کریں تو "2 باقی" بچےگا اس لئے ویری ایبل remaining کی قیمت 2 ہوگی۔

2.5 پروگرامنگ ٹائم (Programming Time)

```
/* This program finds and displays the right most digit of an input
number. */
#include <stdio.h>
void main()
{
      int num, digit;
      printf("Enter a number: ");
      scanf("%d", &num);
      digit = num % 10;
      printf("Right most digit of number you entered is: %d",
      digit);
}
```

Output

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

سرگرمی 2.7:

ایک پروگرام لکھیں جو صارف سے 2 ہندسوں والا نمبر اِن پٹ لے اور ایک ہندسے کو دوسرے ہندسے سے ضرب دے کر آؤٹ پُٹ دکھائے۔

سرگرمی 2.8:

ایک پروگرام لکھیں جو سیکنڈ اِن پٹ لے اور ان کے برابر گھنٹے، منٹ اور سیکنڈ شمار کرے۔

اہم نوٹ:

C- لینگوئج میں ارتھمیٹک سٹیٹمنٹس لکھتے ہوئے ایک عام غلطی الجبرا کے روزمرہ کے اصولوں کو استعمال کرنا ہے۔ مثلاً 6*y کو 6y لکھنا اور x*x*x کو x^3 لکھنا وغیرہ۔ اس کے نتیجے میں کمپائلر ایرر (compiler error) آتا ہے۔

سرگرمی 2.9:

الجبرا کے ایکسپریشنز کو C- ایکپریشنز میں تبدیل کریں۔

$$x = 6y + z$$

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

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

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

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

2.2.3 ریلیشنل آپریٹرز (Relational Operators)

ریلیشنل آپریٹر دو قیمتوں کے تعلق کا تعین کرنے کے لیے ان کا موازنہ کرتے ہیں ریلیشنل آپریٹرز بتاتے ہیں کہ آیا دونوں قیمتیں برابر ہیں یا برابر نہیں ہیں۔ ایک قیمت دوسری سے بڑی ہے یا چھوٹی ہے۔ C- لینگوئج میں ہم ریلیشنل آپریٹرز کو نیومیرک (Numeric) اور Char ٹائپ کے ڈیٹا پر استعمال کر سکتے ہیں ٹیبل 2.3 میں C- لینگوئج کے ریلیشنل آپریٹر اور ان کی تفصیل دی گئی ہے۔

تفصیل	ریلیشنل آپریٹر
کے برابر ہے	==
کے برابر نہیں ہے	!=
سے زیادہ	>
سے کم	<
سے زیادہ یا برابر	>=
سے کم یا برابر	<=

ٹیبل 2.3: بنیادی ریلیشنل آپریٹرز تفصیل کے ساتھ

ریلیشنل آپریٹرز دو آپرینڈز پر آپریشنز انجام دیتے ہیں اور نتیجہ ایک بولین ایکسپریشن (Boolean Expression) کی صورت میں دیتے ہیں یعنی (true or false) صحیح یا غلط۔ اگر صحیح (true) ہو تو قیمت 1 ہوگی اور اگر غلط (false) ہو تو قیمت 0 سے ظاہر کی جائے گی۔ یہ تصوّر اس ٹیبل 2.4 میں واضح کیا گیا ہے۔

جواب	تفصیل	ریلیشنل ایکسپریشن
true	5 برابر ہے 5 کے؟	5==5
true	5 برابر نہیں ہے 7 کے؟	5!=7
false	5 بڑا ہے 7 سے؟	5>7
true	5 چھوٹا ہے 7 سے؟	5<7
true	5 بڑا ہے یا برابر ہے 5 کے؟	5>=5
false	5 چھوٹا ہے یا برابر ہے 4 کے؟	5<=4

ٹیبل 2.4: ریلیشنل آپریٹرز کی مثال کے ساتھ وضاحت

2.2.4 اسائنمنٹ آپریٹر (=) اور برابر کا آپریٹر (==) (Assignment operator(=)&Equal to Operator(==))

C- لینگوئج میں (==) آپریٹر یہ چیک کرنے کے لئے استعمال ہوتا ہے کہ دو ایکسپریشن برابر ہیں یا نہیں جبکہ (=) آپریٹر دائیں طرف والے ایکسپریشن کو بائیں طرف والے ویری ایبل سے منسوب کرتا ہے۔ ڈبل ایکوئل (==) آپریٹر چیک کرتا ہے کہ دونوں طرف یہ آپرینڈز برابر ہیں یا نہیں۔ سنگل ایکوئل (Single Equal) آپریٹر (=) دائیں آپرینڈ کو بائیں طرف والے ویری ایبل سے منسوب کرتا ہے۔

اہم نوٹ:
ہم ریلیشنل ایکسپریشن کا نتیجہ دکھانے کے لئے printf فنکشن بھی استعمال کر سکتے ہیں۔ مثال کے طور پر یہ سٹیٹمنٹس دیکھیں۔

```
printf("%d",5==5);        //This statement dispay 1
printf("%d",5>7);         //This statement dispay 0
```

سرگرمی 2.10:

نیچے دی گئی ایکسپریشنز کا بولین رزلٹ بتائیں۔جبکہ ویری ایبلز اور ان کی قیمتیں یہ ہیں x=3 ، y=7

(x+4)==y	(2+5)>y
(y/2)>=x	x!=(y-4)
(x+3)<=20	-1<x

2.2.5 منطقی آپریٹرز (Logical Operators)

لاجیکل آپریٹرز بولین ایکسپریشنز پر آپریشن سرانجام دیتے ہیں اور جواب بھی ایک بولین ایکسپریشن ہوتا ہے۔جیسا کہ ہم پڑھ چکے ہیں کہ ریلیشنل آپریشن کا جواب بولین ایکسپریشن ہوتا ہے اس لیے لاجیکل آپریٹرز ایک سے زیادہ ریلیشنل ایکسپریشنز کی قیمت نکالنے کے لیے استعمال ہوتے ہیں۔ٹیبل 2.5 میں C-لینگویج کے منطقی آپریٹرز دیے گئے ہیں:

آپریٹر	تفصیل
||	منطقی OR
&&	منطقی AND
!	منطقی NOT

ٹیبل 2.5 بنیادی لاجیکل آپریٹرز اور ان کی تفصیل

AND آپریٹر (&&):

AND آپریٹر (&&) دو بولین ایکسپریشنز بطور آپرینڈ لیتا ہے اور جواب اسی صورت میں true ہوتا ہے اگر دونوں آپرینڈز true ہوں۔اگر کوئی ایک بھی آپرینڈ false ہو تو جواب false ہوتا ہے۔ٹیبل 2.6 میں AND آپریٹر کا ٹروتھ ٹیبل دیا گیا ہے۔

Expression	Result
False && False	False
False && True	False
True && False	False
True && True	True

ٹیبل 2.6: AND آپریٹر کا ٹروتھ ٹیبل

OR آپریٹر (||):

OR آپریٹر بولین ایکسپریشن لیتا ہے اور جواب true ہوتا ہے اگر کوئی ایک آپرینڈ بھی true ہو۔ ٹیبل 2.7 میں OR آپریٹر کا ٹروتھ ٹیبل دیا گیا ہے۔

Expression	Result
False || False	False
False || True	True
True || False	True
True || True	True

ٹیبل 2.7: OR آپریٹر کا ٹروتھ ٹیبل

NOT آپریٹر (!):

NOT آپریٹر بولین ایکسپریشن کی قیمت کو الٹ دیتا ہے یعنی اگر قیمت درست (true) ہو تو غلط (false) کر دیتا ہے اور غلط (false) ہو تو درست (true) کر دیتا ہے۔

Expression	Result
!(True)	False
!(False)	True

ٹیبل 2.8 NOT آپریٹر کا ٹروتھ ٹیبل ہے

لاجیکل آپریٹر کی مثالیں:

ٹیبل 2.9 میں لاجیکل آپریٹر کا تصور مثالوں کی مدد سے واضح کیا گیا ہے۔

لاجیکل ایکسپریشن	تفصیل	جواب
3<4&&7>8	3 چھوٹا ہے 4 سے AND 7 بڑا ہے 8 سے؟	false
3==4||3>1	3 برابر ہے 4 کے OR 3 بڑا ہے ایک سے؟	true
!(4>2||2==2)	NOT(4 بڑا ہے 2 سے OR 2 برابر ہے 2 کے)؟	false
6<=6&&!(1>2)	6 چھوٹا ہے یا برابر ہے 6 کے AND NOT(1 بڑا ہے 2 سے)؟	true
8>9||!(1<=0)	8 بڑا ہے 9 سے OR NOT(1 چھوٹا ہے یا برابر ہے 0 کے)؟	true

ٹیبل 2.9: لاجیکل آپریٹر کی مثال کے ساتھ وضاحت

کیا آپ جانتے تھے؟

C- لینگوئج شارٹ سرکٹ ایویلیوئیشن (Short circuit evaluation) کرتی ہے۔ اس کا مطلب ہے کہ:

1۔ اگر AND اوپریٹر کو حل کرتے ہوئے ایکپریشن آپریٹر کی بائیں طرف والا حصہ false ہو تو پورا ایکسپریشن حل کیے بغیر جواب false ہو جاتا ہے۔

2۔ اگر OR اوپریٹر کو حل کرتے ہوئے ایکپریشن آپریٹر کی بائیں طرف والا حصہ true ہو تو پورا ایکپریشن حل کئے بغیر جواب true ہو جاتا ہے۔

سرگرمی 2.11:

فرض کریں دیے گئے ویری ایبلز کی قیمتیں ہیں x=4, y=7, z=8 جوابی ایکسپریشن بتائیں۔

```
x== 2   ||  y==8          7>= y   &&  z<5
z>=5    ||  x<= -3        y==7    &&  !(true)
x!=y    ||   y<5          !(z>x)
```

2.2.6 یونری بمقابلہ بائنری آپریٹرز (Uniry vs Binary Operators)

وہ تمام آپریٹرز جن کا اس باب میں ذکر کیا گیا ہے۔ آپریٹرز کی تعداد کی بنا پر ان کی بنیادی دو قسمیں ہیں:

1۔ یونری آپریٹرز: یونری آپریٹرز کے لیے صرف ایک آپرینڈ درکار ہوتا ہے مثلاً لاجیکل NOT آپریٹر کا ایک ہی آپرینڈ ہوتا ہے۔ سائن آپریٹر (-) بھی یونری آپریٹر کی ایک مثال ہے۔ جیسے 5-

2۔ بائنری آپریٹرز: بائنری آپریٹر کے لیے دو آپرینڈ درکار ہوتے ہیں مثلاً تمام ارتھمیٹک اور ریلیشنل آپریٹرز ہیں۔ && اور || لاجیکل آپریٹر بھی بائنری آپریٹرز ہیں۔

کیا آپ جانتے تھے؟

C- لینگوئج میں ایک ٹرنری آپریٹر (Ternary Operator) بھی ہوتا ہے جو تین آپرینڈز پر کام کرتا ہے۔

2.2.7 آپریٹرز کی ترجیح (Operators Precedence)

اگر ایک ایکسپریشن میں ایک سے زیادہ آپریٹرز ہوں تو سوال یہ پیدا ہوتا ہے کہ پہلے کسے حل کریں گے۔ اس مسئلے کو حل کرنے کے لیے آپریٹر کی ترجیح ٹیبل 2.10 میں دی گئی ہے۔ جس آپریٹر کی ترجیح زیادہ ہو وہ دوسرے آپریٹر سے پہلے حل کیا جاتا ہے۔ اگر ایک جیسی ترجیح ہو تو دائیں طرف والے آپریٹر سے پہلے بائیں طرف والا آپریٹر حل کریں گے۔

Example:

```
result = 18  / 2 * 3 + 7 % 3 + ( 5 * 4);      // evaluate ( )
result = 18  / 2 * 3 + 7 % 3 + 20;            // evaluate /
result = 9 * 3 + 7 % 3 + 20;                  // evaluate *
result = 27 + 7 % 3 + 20;                     // evaluate %
result = 27 + 1 + 20;                         // evaluate +
result = 28 + 20;                             // evaluate +
result = 48;
```

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

ٹیبل 2.10: آپریٹرز اور ان کی ترجیح

سرگرمی 2.12:

ان ایکسپریشنز کے جوابات لکھیں۔

6 / (5 + 3)
7 + 3 * (12 + 2)
25 % 3 * 4
34 – 9 * 2 / (3 * 3)
18 / (15 – 3 * 2)

خلاصہ:

ہمیں پروگرام لکھتے ہوئے ان پٹ دینے اور آؤٹ پٹ دکھانے کے لیے ایک طریقہ کار کی ضرورت ہوتی ہے۔ ان پٹ / آؤٹ پٹ آپریشنز کے لیے ہر پروگرامنگ لینگوئج کے اپنے کی۔ورڈز (Keywords) یا سٹینڈرڈ بلٹ اِن فنکشنز ہوتے ہیں۔

- **Printf** C-لینگوئج کا ایک بلٹ اِن فنکشن ہے۔ اس کا نام پرنٹ فارمیٹڈ (Print Formatted) سے نکلا ہے اور یہ سکرین پر فارمیٹڈ آؤٹ پٹ دکھانے کے لیے استعمال ہوتا ہے۔
- **فارمیٹ سپیسفائرز** ان پٹ اور آؤٹ پٹ اوپریشنز میں ڈیٹا کا فارمیٹ بتانے کے لیے استعمال ہوتا ہے۔ فارمیٹ سپیسفائر سے پہلے ہمیشہ پرسنٹیج (%) سائن آتا ہے۔
- **scanf** C-لینگوئج کا ایک بلٹ اِن فنکشن ہے جو صارف سے ان پٹ لیتا ہے۔
- **()getch** فنکشن صارف سے ایک کریکٹر اِن پٹ لینے کے لیے استعمال ہوتا ہے۔ اس فنکشن کو صرف کریکٹر ان پٹ دیا جا سکتا ہے۔ صارف جو کریکٹر درج کرتا ہے وہ سکرین پر نظر نہیں آتا۔
- **سٹیٹمنٹ ٹرمینیٹر** کمپائلر کو یہ بتاتا ہے کہ لائن ختم ہو گئی ہے۔ C-لینگوئج میں سیمی کولن (;) بطور سٹیٹمنٹ ٹرمینیٹر استعمال ہوتا ہے۔
- **اسکیپ سیکوئنس** printf کو بتاتا ہے کہ معمول سے ہٹ کر کام کرنا ہے۔ یہ اسکیپ کریکٹر اور ایک کریکٹر جو خاص فنکشنیلٹی سے منسوب ہوتا ہے ان کا مجموعہ ہے۔
- **اسکیپ سیکوئنس** n\ بتاتا ہے کہ کرسر کوئی لائن کے شروع پر لے جانا ہے۔ یہ اسکیپ سیکوئنس آؤٹ پٹ کو ایک سے زیادہ لائنوں پر دکھانے کے لیے استعمال ہوتا ہے۔
- **اسکیپ سیکوئنس** t\ بتاتا ہے کہ کرسر کو افقی طور پر اگلے ٹیب سٹاپ پر لے جانا ہے۔ ایک ٹیب سٹاپ آٹھ سپیسز کا مجموعہ ہوتا ہے۔
- **ارتھمیٹک آپریٹرز**، اسائنمنٹ آپریٹر، ریلیشنل آپریٹر اور لاجیکل آپریٹرز بنیادی آپریٹرز ہیں۔
- ارتھمیٹک آپریٹرز ارتھمیٹک فنکشنز کو حل کرنے کے لیے ڈیٹا پر کیلکولیشنز سر انجام دینے میں استعمال ہوتے ہیں۔ ارتھمیٹک اپریٹرز /، *، -، +، % ہیں۔
- ماڈولس آپریٹر ایک بائنری آپریٹر ہے جو بائیں آپرینڈ کو دائیں آپرینڈ پر تقسیم کرتا ہے اور تقسیم کے بعد بچنے والی رقم جواب ہوتا ہے۔ ماڈولس آپریٹر انٹیجر (integer) ڈیٹا ٹائپس پر کام کرتا ہے۔

- **ریلیشنل اوپریٹرز** دو قیمتوں کے تعلق کا تعین کرنے کے لیے ان کا موازنہ کرتے ہیں تمام ریلیشنل اوپریٹرز بائنری اوپریٹرز ہیں۔اور یہ بائیں طرف سے دائیں طرف کام کرتے ہیں۔
- **لاجیکل اوپریٹرز** بولین ایکسپریشنز پر اوپریشن انجام دیتے ہیں اور جواب ایک بولین ویلیو ہوتی ہے۔
- لاجیکل **AND** اوپریٹر کا جواب true ہوتا ہے اگر اوپریٹر کے دونوں اطراف کے ایکسپریشنز true ہوں جبکہ لاجیکل OR اوپریٹر کا جواب true ہوتا ہے جب دونوں میں سے کوئی بھی ایکسپریشن true ہو۔
- لاجیکل **NOT** اوپریٹر کا جواب true ہوتا ہے اگر ایکسپریشن false ہو اور جواب false ہوتا ہے اگر ایکسپریشن true ہو۔
- پورے ایکسپریشن کو حل کیے بغیر آپریشن کا جواب دینا شارٹ سرکٹنگ (Short circuiting) کہلاتا ہے۔
- **اوپریٹرز** تین قسم کے ہوتے ہیں یونری اوپریٹر، بائنری اوپریٹر اور ٹرنری اوپریٹر جن کو بالترتیب ایک، دو اور تین اوپرینڈز اوپریشن انجام دینے کے لیے درکار ہوتے ہیں۔
- اوپریٹرز کی ترجیح سے پتا چلتا ہے کہ کون سا آپریشن پہلے انجام دینا ہے۔مختلف اوپریٹرز کی مختلف ترجیح ہوتی ہے۔جن کی ترجیح زیادہ ہو انھیں پہلے حل کیا جاتا ہے اور جن کی ترجیح سب سے کم ہو انھیں آخر میں حل کیا جاتا ہے۔

مشق

سوال نمبر 1: کثیر الانتخابی سوالات۔

1۔printf______ قسم کا ڈیٹا پرنٹ کرنے کے لیے استعمال ہوتا ہے۔

الف) int ب) float ج) char د) پہلے تینوں

2۔C scanf- پروگرامنگ لینگوئج میں______ ہے۔

الف) مطلوبہ لفظ ب) لائبریری ج) فنکشن د) کوئی بھی نہیں

3۔getch() صارف سے______ ان پٹ لینے کے لیے استعمال ہوتا ہے۔

الف) int ب) float ج) char د) پہلے تینوں

4۔کوڈ (Code) کا یہ حصہ ایگزیکیوٹ ہونے کے بعد متغیر a کی قیمت کیا ہوگی؟

```
int a = 4;
float b = 2.2;
a = a * b;
```

الف) 8.8 ب) 8 ج) 8.0 د) 8.2

5۔ان میں سے کوڈ کی کونسی لائن صحیح ہے؟

الف) int=20; ب) grade='A' ج) line=this is a line د) کوئی بھی نہیں

6۔ان میں سے کس آپریٹر کی ترجیح سب سے زیادہ ہے؟

الف) / ب) = ج) > د) !

7۔ان میں سے کون سی آپشن آپریٹر کی قسم نہیں ہے؟

الف) ارتھمیٹک آپریٹر ب) ریلیشنل آپریٹر ج) چیک آپریٹر د) لاجیکل آپریٹر

8۔آپریٹر %______ کیلکو لیٹ کرنے کے لیے استعمال ہوتا ہے۔

الف) پرسنٹیج ب) ریمینڈر (بقیہ رقم) ج) فیکٹوریل د) مربع

9۔ان میں سے کون سا کریکٹر C- لینگوئج میں درست ہے۔

الف) "here" ب) "a" ج) 'a' د) کوئی بھی نہیں

10) C لینگوئج کے بارے میں کونسی آپشن درست ہے؟

(الف) C ایک کیس سینسٹو (case Sensitive) لینگوئج نہیں ہے۔
(ب) کی۔ورڈز کو ویری ایبلز کے نام کے طور پر استعمال کیا جاسکتا ہے۔
(ج) تمام لاجیکل آپریٹرز بائنری آپریٹرز ہوتے ہیں۔
(د) کوئی بھی نہیں۔

سوال نمبر2: غلط درست کا انتخاب کریں۔

1) ایک انٹیجر کی قیمت زیادہ سے زیادہ 32000 ہو سکتی ہے۔ درست/غلط
2) فارمیٹ سپیسفائرز کے شروع میں % سائن آتا ہے۔ درست/غلط
3) تقسیم کے آپریٹر کی ترجیح ضرب کے آپریٹر سے زیادہ ہے۔ درست/غلط
4) ()getch صارف سے ہر قسم کا ڈیٹا ان پُٹ لینے کے لیے استعمال ہوتا ہے۔ درست/غلط
5) scanf آؤٹ پُٹ آپریشنز کے لیے استعمال ہوتا ہے۔ درست/غلط

سوال نمبر3: درج ذیل کی تعریف لکھیں۔

(1) سٹیٹمنٹ ٹرمینیٹر (2) فارمیٹ سپیسفائر (3) اسکیپ سیکوئنس

(4) Scan f (5) ماڈولس آپریٹر

سوال نمبر4: ان سوالوں کے مختصر جواب دیں۔

1) Scanf اور ()getch میں کیا فرق ہے؟
2) C-لینگوئج کا کونسا فنکشن سکرین پر آؤٹ پُٹ دکھانے کے لیے استعمال ہوتا ہے؟
3) ان پُٹ آؤٹ پُٹ آپریشنز میں فارمیٹ سپیسفائرز بتانا کیوں ضروری ہے؟
4) اسکیپ سیکوئنسز کیا ہوتے ہیں؟ ہمیں ان کی ضرورت کیوں ہوتی ہے؟
5) ارتھمیٹک آپریشنز میں کونسے آپریٹر استعمال ہوتے ہیں؟
6) ریلیشنل آپریٹرز کیا ہیں؟ مثال دے کر وضاحت کریں۔
7) لاجیکل آپریٹرز کیا ہیں؟ مثال دے کر وضاحت کریں۔
8) یونری آپریٹرز اور بائنری آپریٹرز میں کیا فرق ہے؟
9) == آپریٹرز اور = آپریٹرز میں کیا فرق ہے؟
10) آپریٹرز کی ترجیح سے کیا مراد ہے؟ C-لینگوئج میں کس آپریٹر کی ترجیح سب سے زیادہ ہے؟

سوال نمبر5: کوڈ کے ان حصوں کی آؤٹ پُٹ لکھیں۔

a)
```
# include<stdio.h>
void main()
{
        int x = 2, y = 3, z = 6;
        int ans1, ans2 , ans3;
        s1 = x / z * y ;
        ans2 = y + z / y * 2;
        ans3 = z / x + x * y;
        printf("%d %d %d", ans1, ans2, ans3 );
}
```

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

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

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

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

سوال نمبر 6: کوڈ کے درج ذیل حصوں میں ایررز تلاش کریں۔

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

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;
]
```

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

d)
```
include<stdio.h>
int main ()
{
        float f;
        printf["Enter value: "];
        scanf("%c", &f);
}
```

پروگرامنگ کی مشقیں

مشق 1:

ایک کمپنی میں تنخواہ کا تعین کرنے کا طریقہ کار نیچے دیا گیا 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 جو ملازم کے بنیادی کام کرنے کے گھنٹے اور اضافی گھنٹے بطور ان پُٹ لے، کل تنخواہ کا حساب لگائے اور اسے سکرین پر دکھائے۔

مشق 2:

ایک پروگرام لکھیں جو سینٹی گریڈ میں درجہ حرارت کو ان پُٹ لے اور اسے فارن ہائیٹ میں تبدیل کر کے آؤٹ پُٹ دکھائے۔

درجہ حرارت کو سینٹی گریڈ سے فارن ہائیٹ میں کرنے کا فارمولا یہ ہے۔

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

مشق 3:

ایک پروگرام لکھیں جو ایک printf سٹیٹمنٹ کو استعمال کرتے ہوئے یہ آؤٹ پُٹ دکھائے۔

```
*    *    *    *
1    2    3    4
```

مشق 4:

ایک پروگرام لکھیں جو ایک printf سٹیٹمنٹ کو استعمال کرتے ہوئے یہ آؤٹ پُٹ دکھائے

```
I am a Boy
I live in Pakistan
I am a proud Pakistani
```

مشق 5:

کپڑے کا ایک برینڈ ہر چیز پر 15% ڈسکاؤنٹ دیتا ہے۔ایک خاتون اس برینڈ کی پانچ قمیصیں خریدتی ہے۔ایک پروگرام لکھیں جو ڈسکاؤنٹ کے بعد کی کل قیمت اور ڈسکاؤنٹ کی گئی رقم کا حساب لگائے۔قمیصوں کی اصل قیمتیں یہ ہیں۔

قمیص1 =423

قمیص2 = 320

قمیص3 = 270

قمیص4 = 680

قمیص5 = 520

نوٹ: قمیصوں کی قیمتیں 5 ویری ایبلز میں محفوظ کریں۔

مشق 6:

ایک پروگرام لکھیں جو دو انٹیجر ویری ایبلز کی قیمتوں کا تیسرے ویری ایبل کے استعمال کے بغیر تبادلہ (Swap) کرے۔

مشق 7:

ایک پروگرام لکھیں جو 5 ہندسوں والا نمبر بطور ان پُٹ لے اور پہلا اور آخری ہندسہ جمع کر کے جواب پرنٹ کرے۔

مشق 8:

ایک پروگرام لکھیں جو ماہانہ تنخواہ اور ماہانہ اخراجات جیسے بجلی کا بل، گیس کا بل، کھانے کے اخراجات صارف سے بطور ان پُٹ لے۔ پروگرام ان سب کو کیلکولیٹ کرے۔

کل ماہانہ اخراجات

کل سالانہ اخراجات

ماہانہ بچت

سالانہ بچت

ایک مہینے کی اوسط بچت

ایک مہینے کے اوسط اخراجات

مشق 9:

ایک پروگرام لکھیں جو ایک کریکٹر اور سٹیپس (Steps) کی تعداد صارف سے ان پُٹ لیں۔

پروگرام پھر اس کریکٹر سے اتنے سٹیپس آگے جمپ کرے۔

SAMPLE OUTPUT

Enter Character : a

Enter Step : 2

New Character : C

مشق 10:

ایک پروگرام لکھیں جو دائرے کا رداس (Radius) ان پُٹ لے۔ پروگرام دائرے کا رقبہ شمار کر کے سکرین پر دکھائے۔