

CS

PRACTICAL NO 1

Aim: Program to understand the basic datatype and input/output

Program 1 : Area of rectangle

Algorithm:

Step 1:

Step 2: Initialize three variables for length, breadth and area.

Step 3: Take a input from the user and store the value in the variable declared.

Step 4: find the area of rectangle.

Step 5: Print the area of rectangle.

Step 6: End

Code:

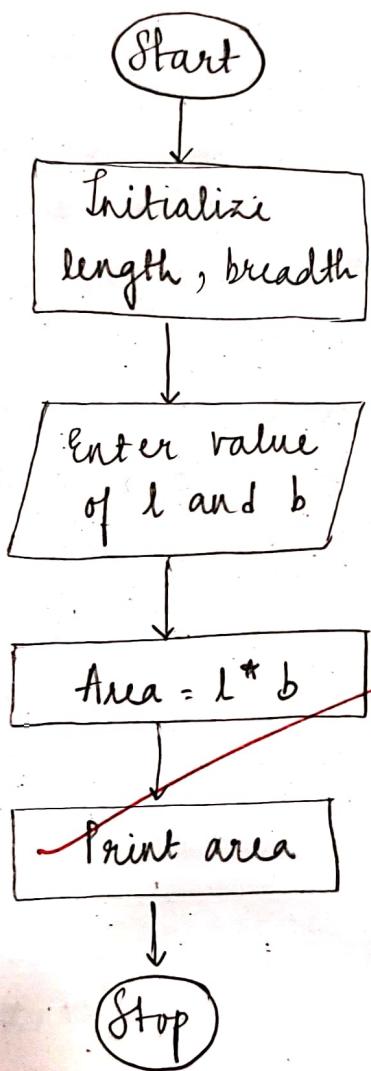
```
#include <stdio.h>
#include <conio.h>
void main()
{
    int l, b, area;
    clrscr();
    printf("Enter length and breadth");
    scanf("%d %d", &l, &b);
    area = l * b;
    printf("The area of rectangle %.d", area);
    getch();
}
```

Output:

Enter length and breadth 7 3

The area of rectangle 21

flowchart:

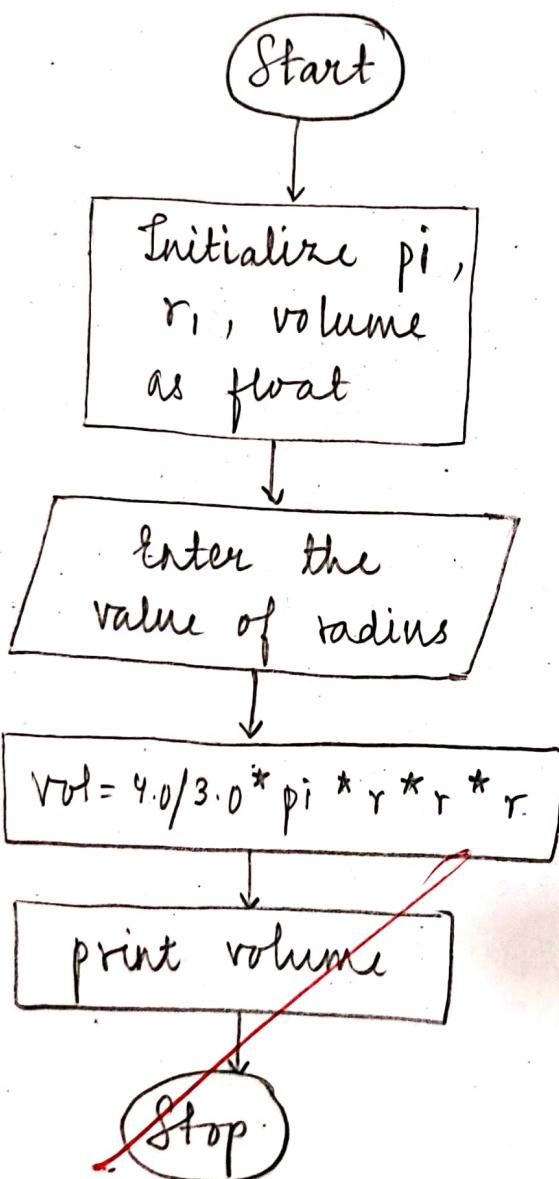


Output:

Enter the radius 7

The volume of sphere 1436.026733

flowchart:



Program 2 : Volume of sphere

Algorithm:

Step 1:

Step 2: Initialize three variables for volume of sphere with datatype.

Step 3: Assign value for pi= 3.14.

Step 4: Clear the screen

Step 5: Take a input from the user with datatype float.

Step 6: Store the value in a variable declared.

Step 7: Perform the operations for volume of sphere.

Step 8: Print the volume of sphere.

Step 9: End

Code:

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

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

```
void main ()
```

```
{
```

```
float pi, r, vol;
```

```
pi = 3.14;
```

```
clrscr();
```

```
printf ("Enter the radius");
```

```
scanf ("%f", &r);
```

```
vol = 40 / 30 * pi * r * r * r;
```

```
printf ("The volume of sphere:", vol);
```

```
getch();
```

```
}
```

Q3.

Program 3: Average of three numbers

Algorithm:

Step 1:

Step 2: Initialize three variables x, y, z and avg for finding the average of numbers.

Step 3: Clear the screen.

Step 4: Take three input from the user and store the values to the given variables.

Step 5: Perform the operations for average of three numbers.

Step 6: Print the average of three numbers.

Step 7: End

Code:

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

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

```
void main ()
```

```
{
```

```
float x, y, z, avg;
```

```
clrscr();
```

```
printf ("Enter three numbers:");
```

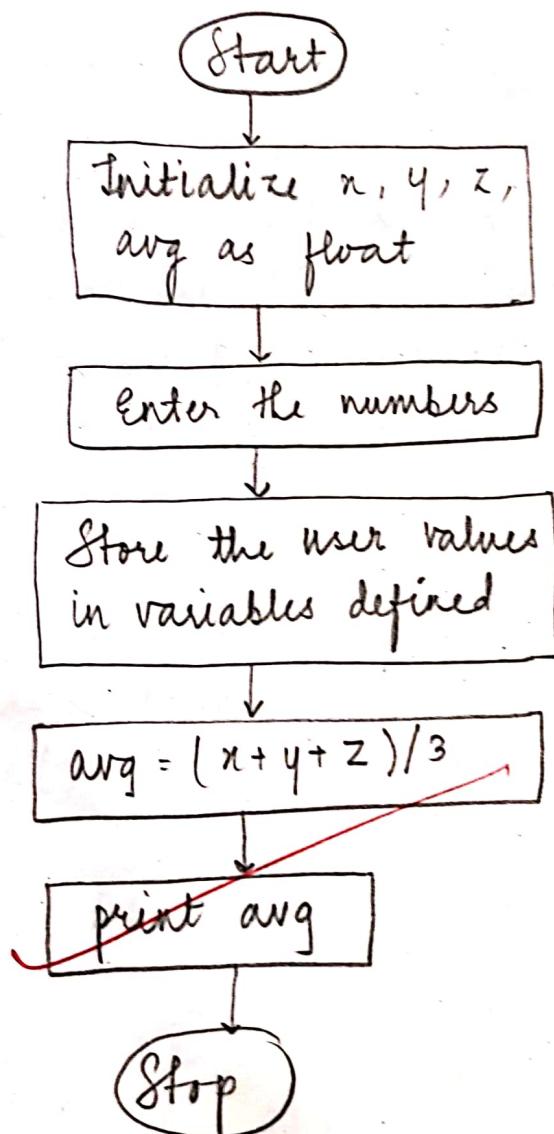
~~```
scanf ("%f %f %f", &x, &y, &z);
```~~~~```
avg = (x+y+z)/3
```~~~~```
printf ("The average of three numbers: ", avg);
```~~~~```
getch();
```~~

```
}
```

Output:

Enter three numbers 3 10 50
The average of three numbers 21

Flowchart:



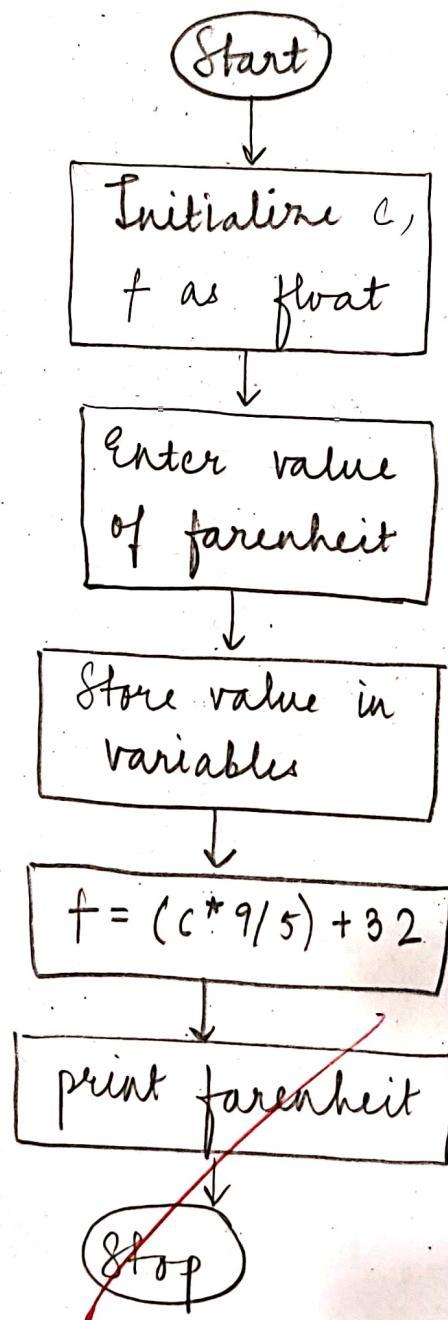
Q8

Output:

Enter the value of celsius 31°

The farenheit is 87.8

Flowchart:



Program 4: Convert temperature from celsius to farenheit.

Algorithm:

Step 1:

Step 2: Initialize two variables for celsius and farenheit.

Step 3: Clear the screen.

Step 4: Take the input from the user of celsius.

Step 5: Convert the celsius to farenheit using this operation $f = (c * 9/5) + 32$

Step 6: Print the farenheit value.

Step 7: End

Code:

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

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

```
void main()
```

```
{
```

```
float c, f;
```

```
clrscr();
```

```
printf("Enter the value of celsius ");
```

```
scanf("%f", &c);
```

```
f = (c * 9/5) + 32;
```

```
printf("The farenheit is : ", f);
```

```
getch();
```

```
}
```

(User input for value of celsius) + twig
 $(f = (c * 9/5) + 32)$

$(f = (c * 9/5) + 32)$ + twig

$(f = (c * 9/5) + 32)$ + twig

Scanned with CamScanner

Program 5: Convert the temperature from farenheit to celsius.

Algorithm:

- Step 1: Initialize two variables with suitable datatype for celsius and farenheit.
- Step 2: Initialize two variables with suitable datatype for celsius and farenheit.
- Step 3: Clear the screen
- Step 4: Take input from the user for farenheit
- Step 5: Store the value of farenheit which is initialized at Step 2
- Step 6: Perform the celsius operation to find the celsius
- $$C = (5.0 / 9.0) * (f - 32)$$
- Step 7: Print the celsius.
- Step 8: End

Code:

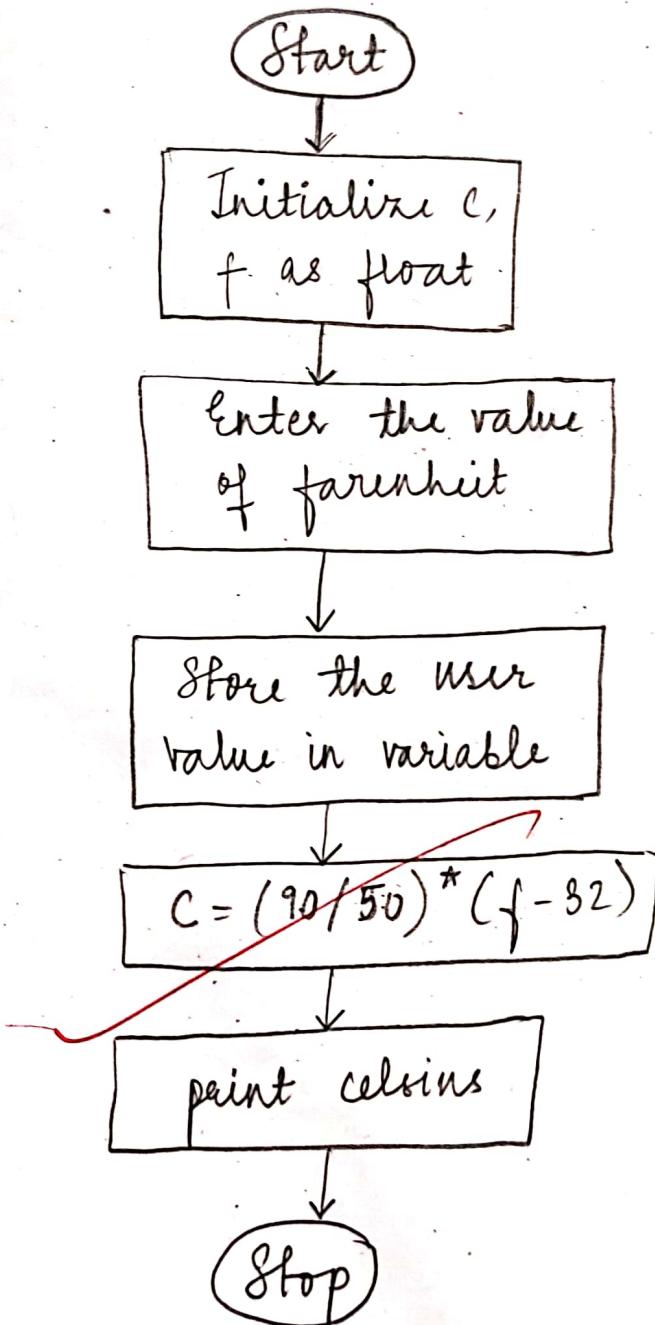
```
#include <stdio.h>
#include <conio.h>
void main()
{
    float c, f;
    clrscr();
    printf("Enter the value of farenheit");
    scanf("%f", &f);
    C = (9.0 / 5.0) * (f - 32);
    printf("The celsius is : ", c);
    getch();
}
```

Output:

enter the value of farenheit 80
The celsius is 26.666667

32

flowchart:



80
26.666667

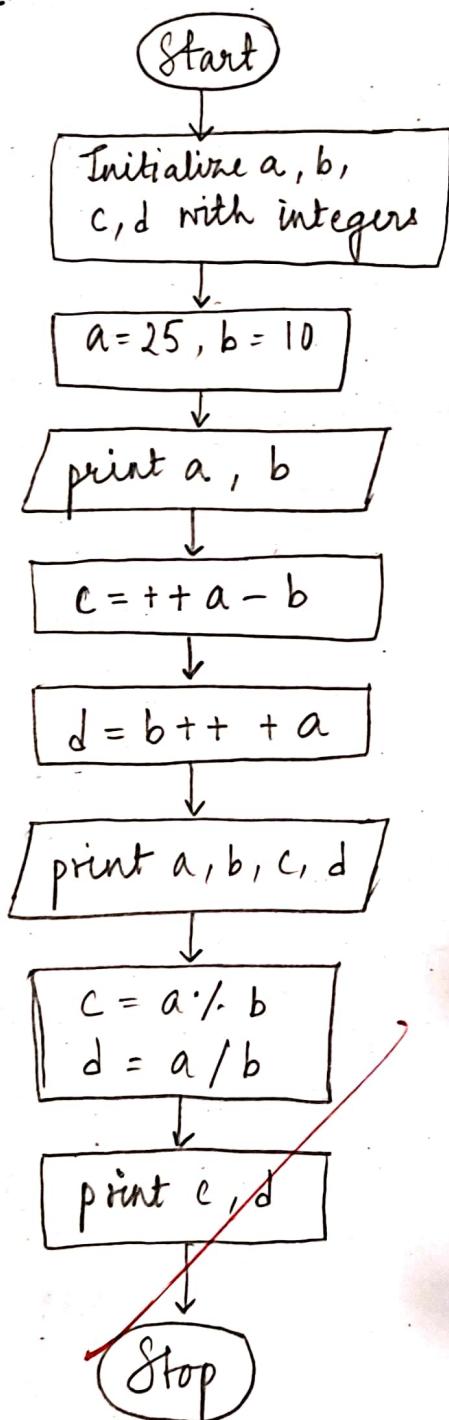
Output:

$a = 25, b = 10$

$a = 26, b = 11, c = 16, d = 36$

$c = 4, d = 2$

Flowchart:



PRACTICAL NO 2

33

Aim: Programs on operators and expressions

Program 1:

Algorithm:

- Step 1: Initialize four variables with datatype int
- Step 2: Clear the screen
- Step 3: Store the value 25 in a and 10 in b
- Step 4: find the value of $a \& b$.
- Step 5: Do the expression $c = ++a - b$
- Step 6: Do post increment b and add to a, store it in d
- Step 7: Print the value of a, b, c, d.
- Step 8: Do $a \% b$ and store in c
- Step 9: Do a / b and store in d
- Step 10: Print the value of c and d

```
# include <stdio.h>
# include <conio.h>
void main()
{
```

```
    int a , b , c , d ;
```

```
    clrscr () ;
```

```
    a = 25 , b = 10 ;
```

```
    printf (" \n a = % .d , b = % .d " , a , b ) ;
```

```
    c = ++a - b ;
```

```
    d = b + +a ;
```

```
    printf (" \n a = % .d , b = % .d , c = % .d , d = % .d " ) ;
```

```
    getch () ;
```

$a, b, c, d);$
 $c = a \% b$
 $d = a / b$
 $\text{printf} (" \ln c=%d, d=%d ", c, d);$
 $\text{getch}();$

Program 2:

Algorithm:

- Step 1: Initialize variable a, b, c with value $a = 8, b = 15,$
 $b \% c = 3$ and x, y, z
- Step 2: Print the value of a, b, c
- Step 3: Perform $a - b / 3 + c^{*(2-1)}$ and store in x
- Step 4: Perform $a - (b / (3+c)^*(2-1))$ and store in y
- Step 5: Perform $a - (b / (3+c)^*2) - 1$ and store in z
- Step 6: Print the value of x, y, z

```

#include <stdio.h>
#include <conio.h>
void main()
{

```

float a, b, c;

$a = 8;$

$b = 15;$

$c = 3;$

$\text{printf} (" \ln a=%f, b=%f, c=%f ", a, b, c);$

$x = a - b / 3 + c^{*(2-1)};$

$y = a - b / (3+c)^*(2-1);$

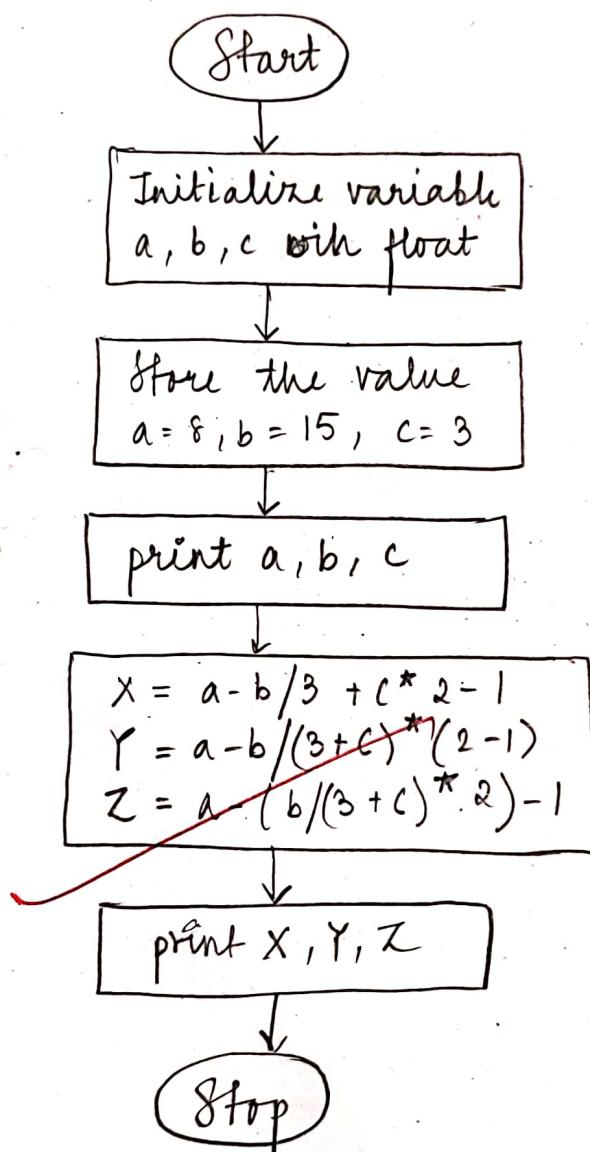
$z = a - (b / (3+c)^*2) - 1;$

Output:

34

$a = 8.000000, b = 15.000000, c = 3.000000$
 $x = 8.000000, y = 5.500000, z = 2.000000$

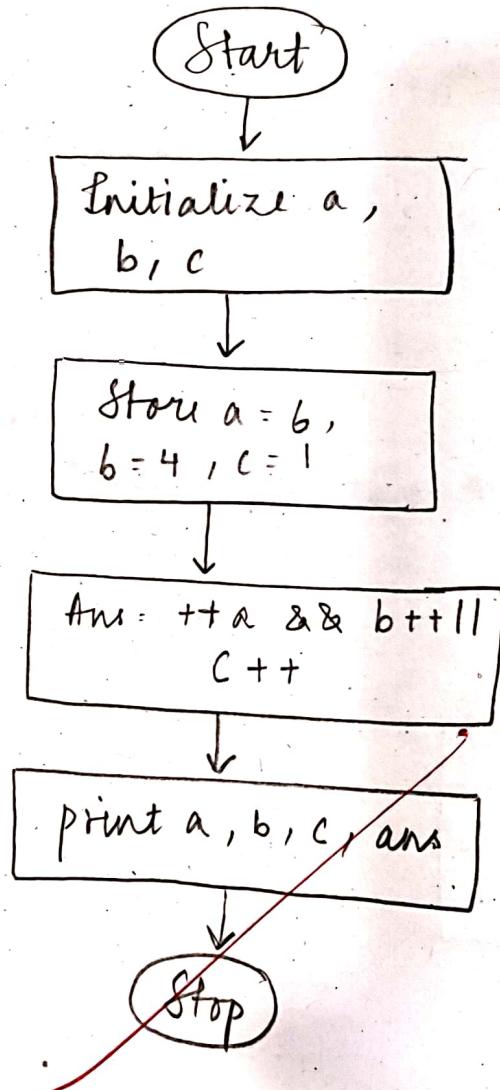
flowchart:



Output :

$$\overline{a=7, b=5, c=1} \quad \text{ans} = 1.000000$$

Flowchart :



```
printf ("In x=%f, y=%f, z=%f", x, y, z);
getch();
```

Program 3 :

Algorithm: ~~Algorithm~~ for ~~value~~ of ~~variable~~ ~~int~~ ~~datatype~~

Step 1: Initialize a, b, c, ans. with datatype integer.

Step 2: Clear the screen.

Step 3: Store the value in a=6, b=4, c=1.

Step 4: Perform expression $a++ \& b++ \parallel c++$ and store the value in ans.

Step 5: Print the value for a, b, c, ans.

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

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

```
void main ()
```

```
{
```

```
int a, b, c, ans;
```

```
clrscr();
```

```
a = 6; b = 4; c = 1;
```

```
ans = ++a  $\&&$  b++  $\parallel$  c++;
```

```
printf ("a=%d, b=%d, c=%d, ans=%d", a, b, c, ans);
```

```
getch();
```

~~Program 4: (a) #include <stdio.h> int main() {~~

~~int a, b, c, x;~~

Algorithm:

- Step 1: Initialize variables a, b, c, & with datatype int.
- Step 2: Clear the screen.
- Step 3: Store x = 10
- Step 4: Post increment the value of x and store in a variable
- Step 5: Decrement the value of x and store in b.
- Step 6: Perform $x++ * --b$ and store in c with value
- Step 7: Print the value of a, b, c, x
- Step 8: End

include <stdio.h>

include <conio.h>

void main()

{

int a, b, c, x;

clrscr();

x = 10;

a = x++;

b = --x;

c = x++ * --b

printf("a=%d, b=%d, c=%d, x=%d", a, b, c);

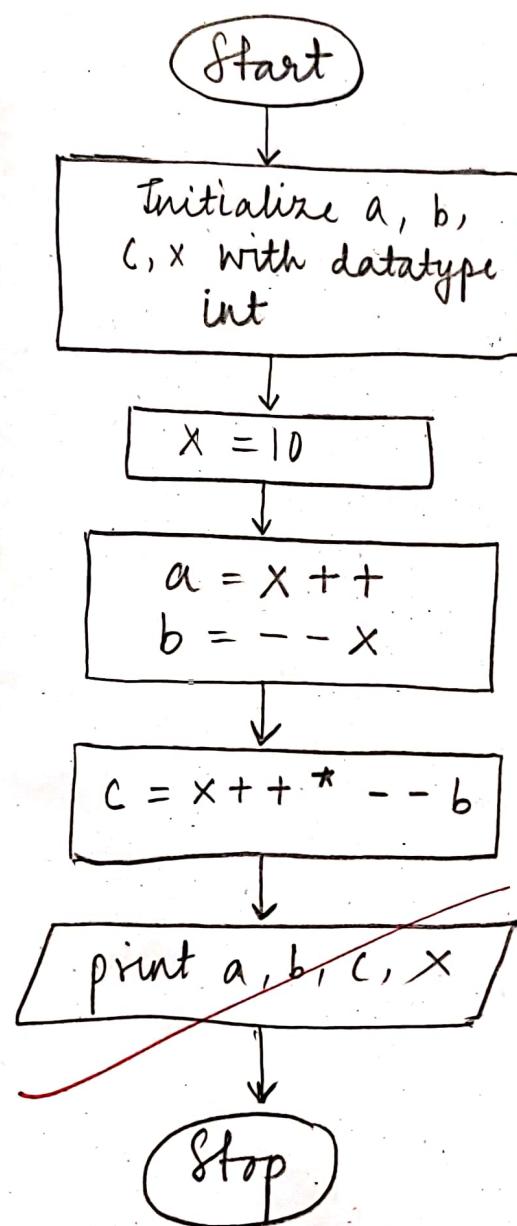
getch();

}

Output:

$a = 10, b = 9, c = 90, x = 11$

flowchart :



$\begin{array}{l} a \\ b \\ c \\ x \end{array} = \begin{array}{l} 10 \\ 9 \\ 90 \\ 11 \end{array}$

36

Output:

- 2
- 4
- 6
- 8
- 10
- 12
- 14
- 16
- 18
- 20

$$11 - x \cdot 0.5 = 2.5$$

Ques: Program to understand looping statements

Program 1: Program to print even numbers from 1 to 100

```
# include <stdio.h>
# include <conio.h>
void main()
{
    int i;
    clrscr();
    for (i=2; i<=20; i=i+2)
    {
        printf ("%d \t", i);
    }
    getch();
}
```

Program 2:

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

Output:

|
| 2
| 2 3
| 2 3 4
| 2 3 4 5

~~88
88~~

Output:

Enter the value of n 10

The sum of all odd nos are 25

Program 3:

```
# include <stdio.h>
# include <conio.h>
void main()
{
    int i, n, sum, x;
    clrscr();
    printf("Enter the value of n ");
    scanf("%d", &n);
    i = 1
    sum = 0;
    do
    {
        if (n == 1)
        {
            sum = sum + i;
        }
        i++;
    }
    while (i <= n)
    printf("The sum of all odd nos are %d", sum);
    getch();
}
```

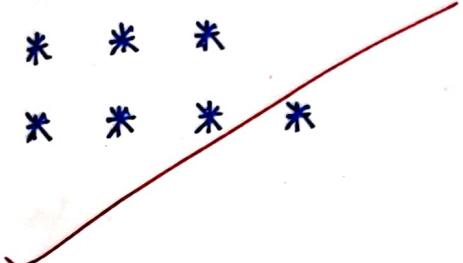
Q.E

Program 4:

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

Output:

*
* *
* * *
* * * *
* * * *



Output:

1
1
2
3
5
8
13
21
34
65
89
144
233
377
610
987
1597
2584

Program 5:

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    int a, b, f, i;
    clrscr();
    a = i;
    b = 0;
    for (i=3; i<=20; i++)
    {
        f = a + b;
        printf ("%d", f);
        a = b;
        b = f;
    }
    getch();
}
```

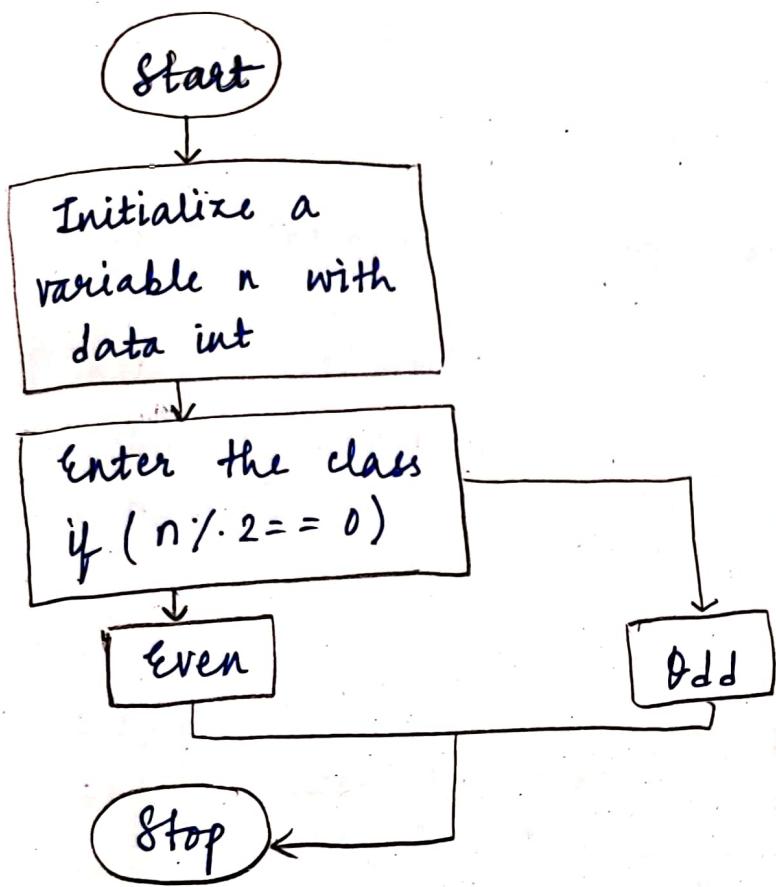
~~8
03/03~~

PRACTICAL NO 3

Aim: Program on decision making and branching

Program 1: Check whether number is odd or even.

```
# include <stdio.h>
# include <conio.h>
void main()
{
    clrscr();
    int n, r;
    printf ("\nEnter value of n : ");
    scanf ("%d", &n);
    r = n % 2;
    if (r == 0)
        printf ("\n %d is even", n);
    else
        printf ("\n %d is odd", n);
    getch();
}
```



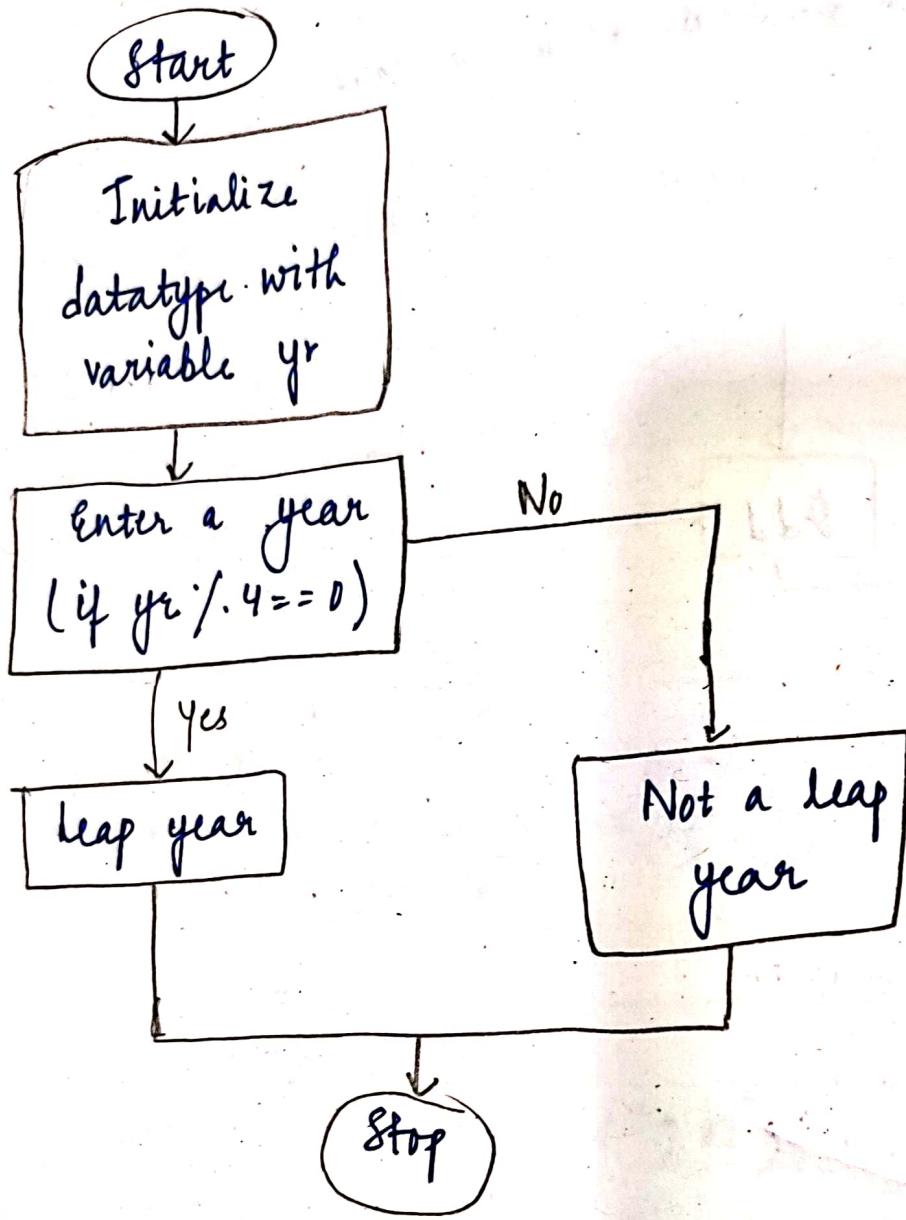
Output:

Enter value of n: 12

12 is even

Enter value of n: 51

51 is odd



Output:

Enter the year 2001

2001 is not a leap year

Enter the year 2004

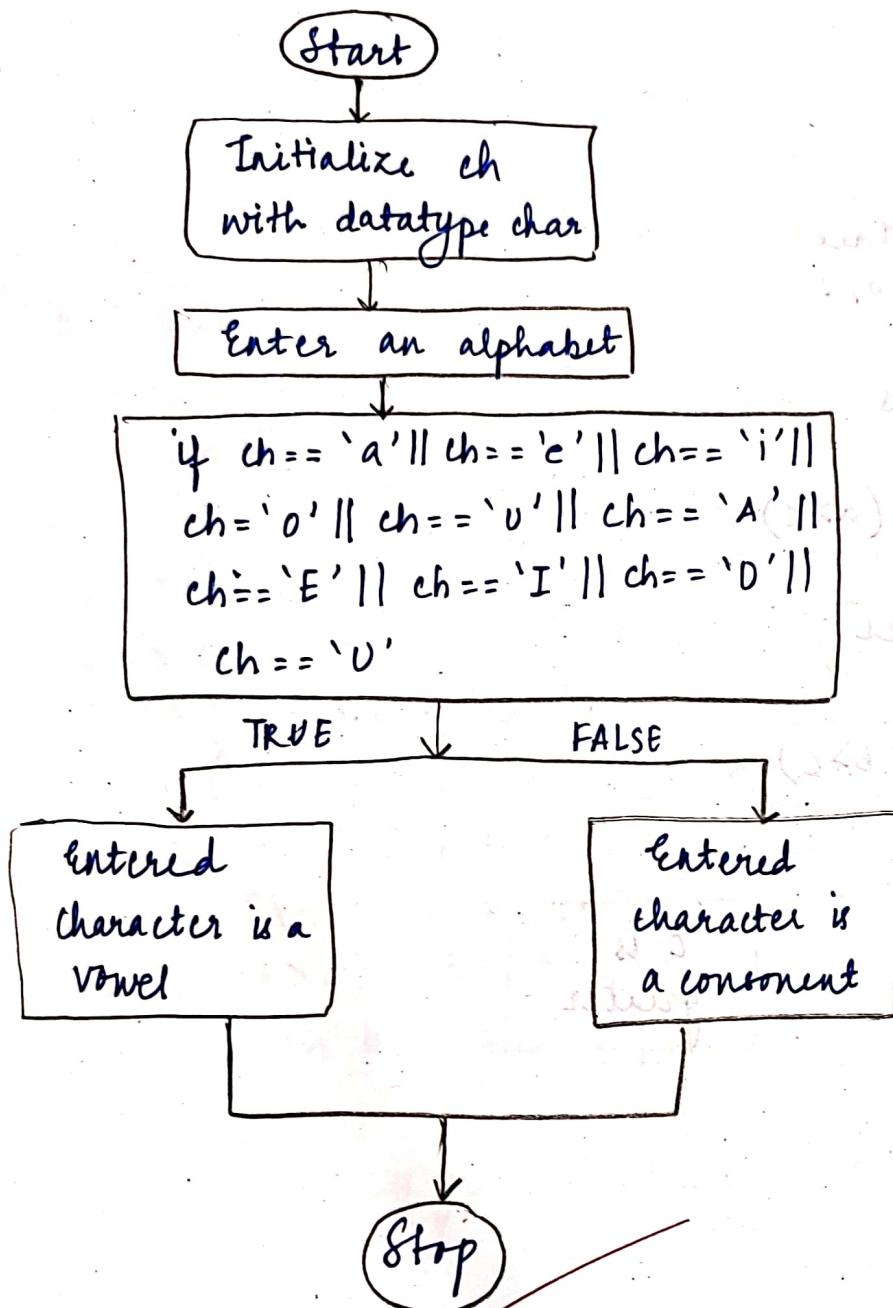
2004 is a leap year

Program 2: Check if the entered year is a leap year or not

```
#include <stdio.h>
#include <conio.h>
void main()
{
    clrscr();
    int r, f;
    printf ("Enter the year");
    scanf ("%d", &y);
    r = y % 4;
    if (r == 0)
        printf ("\n %d is a leap year", y);
    else
        printf ("It is not leap year");
    getch();
}
```

Program 3: Check whether entered alphabet is a vowel or consonant.

```
# include <stdio.h>
# include <conio.h>
void main()
{
    clrscr();
    char ch;
    printf ("\n Enter the alphabet")
    ch = getch();
    if (char == 'a' || ch == "e" || ch == 'i' || ch == 'o' || ch ==
        || ch == 'A' || ch == "E" || ch == 'I' || ch == 'O' || ch ==
    {
        printf (" \n %.c is vowel", ch);
    }
    else
        printf (" \n %.c is a consonant", ch);
    getch();
}
```



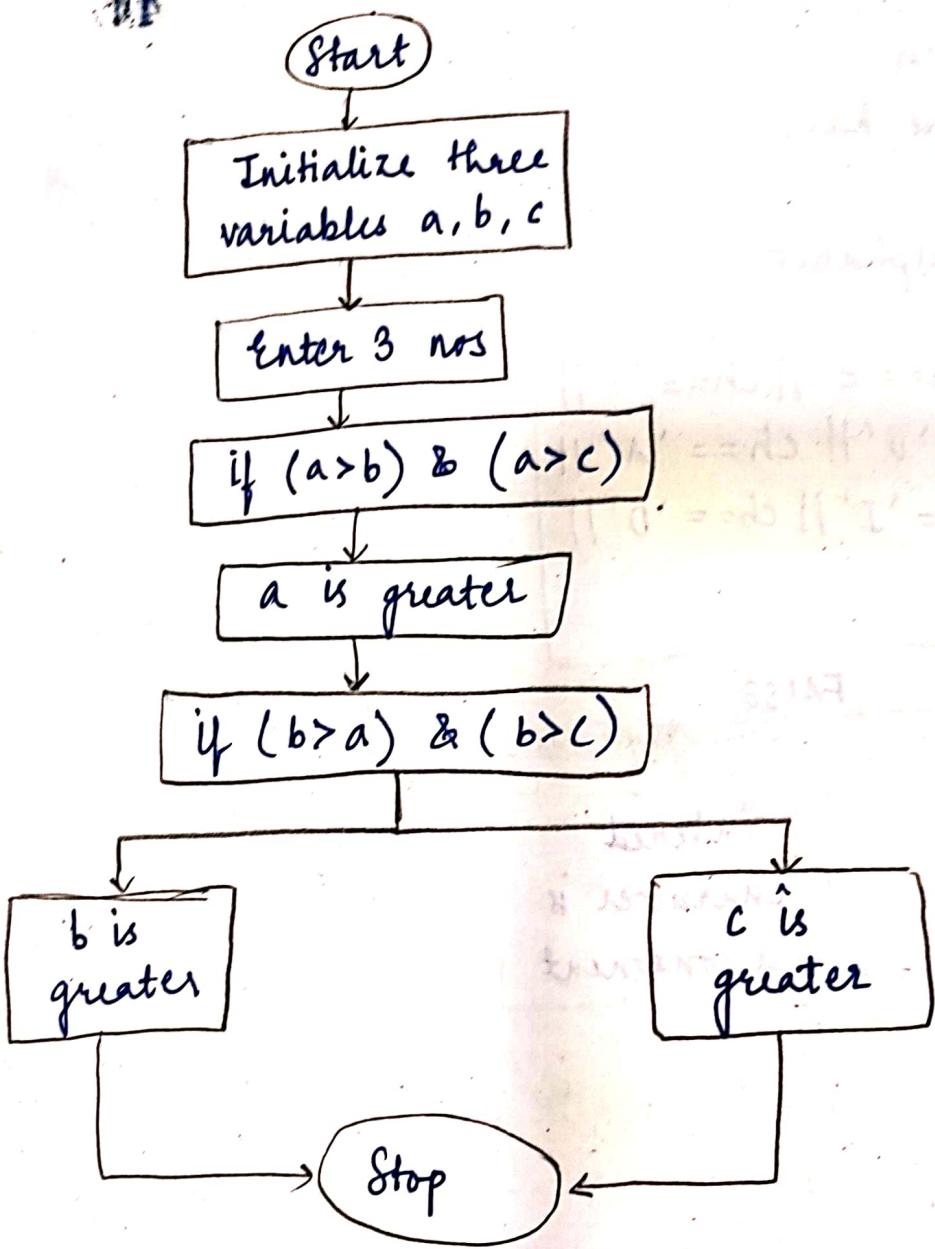
Output:

Enter the alphabet: i

i is a vowel

Enter the alphabet: S

S is a consonant



Output:

Enter 3 nos

3

7

1

b is greater

Program 4: It handles right sign errors at margin of error.

#include <stdio.h>

#include <conio.h>

void main ()

3

int a, b, c;

user () ;

```
printf ("Enter 3 nos");
```

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

if ((a > b) && (a > c))

```
printf ("\\n a is greater");
```

else if ((b > a) && (b > c))

PHM

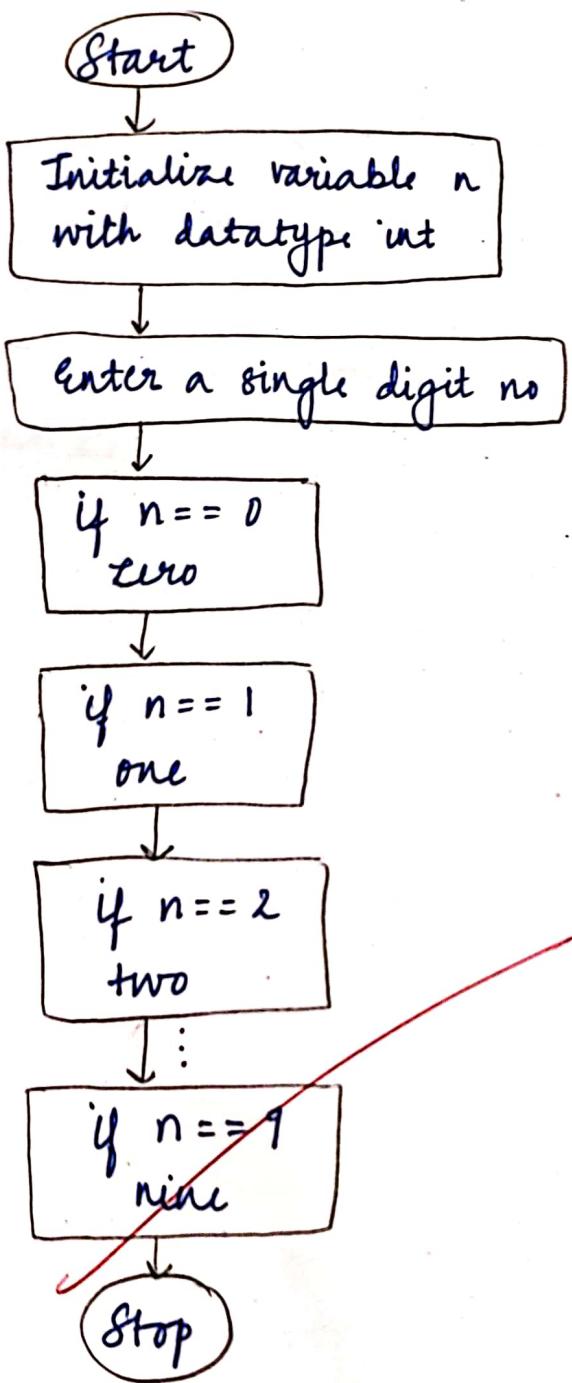
~~else
printf ("\\n c is greater");~~

getch();

3

Program 5: Program to enter single digit decimal number
keyword & print that digit in word form

```
# include <stdio.h>
# include <conio.h>
void main()
{
    clrscr();
    int n
    printf ("\n Enter single digit decimal no: ");
    scanf ("%d", &n);
    if (n == 0)
        printf ("\n zero");
    else if (n == 1)
        printf ("\n one");
    else if (n == 2)
        printf ("\n two");
    else if (n == 3)
        printf ("\n three");
    else if (n == 4)
        printf ("\n four");
    else if (n == 5)
        printf ("\n five");
    else if (n == 6)
        printf ("\n six");
    else if (n == 7)
        printf ("\n seven");
    else if (n == 8)
        printf ("\n eight");
```



```
else if (n == 9)
    printf ("In nine");
```

```
else
```

```
printf ("In error");
```

```
getch();
```

03/03

(A.03/03) shell32.dll

(C.03/03) shell32.dll

(D.03/03)

if (n <= 0 || n > 9) {
 printf ("In error");

else if (n == 1) {
 printf ("One");

else if (n == 2) {
 printf ("Two");

else if (n == 3) {
 printf ("Three");

else if (n == 4) {
 printf ("Four");

else if (n == 5) {
 printf ("Five");

else if (n == 6) {
 printf ("Six");

else if (n == 7) {
 printf ("Seven");

else if (n == 8) {
 printf ("Eight");

else if (n == 9) {
 printf ("Nine");

else {
 printf ("error");

}

PRACTICAL NO 5

Aim: To understand the concept of arrays

⇒ ONE DIMENSIONAL ARRAYS:

Program 1: Find the largest number in an array of numbers.

```
# include <stdio.h>
# include <conio.h>
void main()
{
    clrscr();
    int i, l, a[10];
    printf ("\n Enter 10 data of array : ");
    for ( i=0 ; i<10 ; i++ )
    {
        scanf ("%d", &a[i]);
    }
    l = a[0];
    for ( i=1 ; i<10 ; i++ )
    {
        if ( l < a[i] )
        {
            l = a[i];
        }
    }
    printf ("\n largest : %d", l);
    getch();
}
```

Output:
enter 10
1 2 3
largest

Output:

enter 10 data of array:

1 2 3 4 5 6 7 8 9 10 11

largest = 11

8A:

OUTPUT:

Enter data of array

12 56 90 7 11 46 77 60 10 134

Smallest: 7

Program 2: find the smallest number in an array of 10 elements.

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

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

```
void main()
```

```
{
```

```
clrscr();
```

```
int i, s, a[10];
```

```
printf ("In Enter 10 data of array: ");
```

```
for ( i=0; i<10; i++)
```

```
{
```

```
scanf ("%d", &a[i]);
```

```
}
```

```
s = a[0];
```

```
for ( i=1; i<10; i++)
```

```
{
```

```
if ( s > a[i] )
```

```
{
```

```
    s = a[i];
```

```
}
```

```
}
```

```
printf ("Smallest : %d ", s);
```

```
getch();
```

```
}
```

E.P

Program 3: find number of positive numbers present in array of 10 data:

```
# include <stdio.h>
# include <conio.h>
void main()
{
    clrscr();
    int i, c, a[10];
    printf("\n Enter 10 data of array :");
    for (i=0; i<10; i++)
    {
        scanf("%d", &a[i]);
    }
    c = 0;
    for (i=0; i<10; i++)
    {
        if (a[i]>0)
        {
            c = c + 1;
        }
    }
    printf("\n No of positive no = %d", c);
    getch();
}
```

OUTPUT
Enter
1 2
No of positive no = 2

INPUT :

Enter 10 data of array

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

No of positive no = 6

Q2

OUTPUT:

Enter 10 data of array

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

No of negative no = 5

Program 4: Find number of negative numbers present in an array of 10 data.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    clrscr();
    printf("In Enter 10 data of array :");
    for(i=0; i<10; i++)
    {
        scanf("%d", &a[i]);
    }
    c=0;
    for(i=0; i<10; i++)
    {
        if(a[i] < 0)
        {
            c=c+1;
        }
    }
    printf("In No of negative no = %d", c);
    getch();
}
```

16.

Program 5: Write a program to find numbers of odd numbers present in array of 10 data:

```
# include <stdio.h>
# include <conio.h>
void main()
{
    int i, r, c, a[10];
    clrscr();
    printf("In Enter 10 data of array:");
    for (i=0; i<10; i++)
    {
        scanf("%d", &a[i]);
    }
    c = 0;
    for (i=0; i<10; i++)
    {
        r = a[i] % 2;
        if (r == 1)
            c++;
    }
    printf("In No of odd no: %d", c);
    getch();
}
```

OUTPUT

Enter 10 data of array

1 2 3 4 5 6 7 8 9 10

No of odd no = 5

23

OUTPUT

Enter 10 data of array

1 2 3 4 5 6 7 8 9 10

No of even no = 5

Program 6: find no of even numbers

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, r, c, a[10];
    clrscr();
    printf ("In Enter 10 data of array");
    for (i=0; i<10; i++)
    {
        printf ("%d", &a[i]);
    }
    c = 0;
    for (i=0; i<10; i++)
    {
        r = a[i] % 2;
        if (r == 0)
            c++;
    }
    printf ("In No of even no. = %d", c);
    getch();
}
```

32
Program 7: Write a program to find average of 10 elements of array

```
# include <stdio.h>
# include <conio.h>
void main ()
{
    clrscr();
    int a[10], i, s;
    float avg;
    printf ("Enter 10 data of array: ");
    for (i=0; i<10; i++)
    {
        scanf ("%d", &a[i]);
    }
    s = 0;
    for (i=0; i<10; i++)
    {
        s = s + a[i];
    }
    avg = s / 10;
    printf ("\n SUM = %d", s);
    printf ("\n AVERAGE = %.2f", avg);
    getch();
}
```

8

OUTPUT

Enter 10 data of array:

1 2 3 4 5 6 7 8 9 10

sum = 55

average = 5.500000

~~83~~
03/03

12

Output:

Enter your name: Abhay

My name is : Abhay

Aim: Program to understand string manipulation

Program 1: Write a program to display your name using string.

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

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

```
void main ()
```

```
{
```

```
clrscr();
```

```
char name [20];
```

```
printf ("Enter your name");
```

```
scanf ("%s", name);
```

```
printf ("\n My name is : %s", name);
```

```
getch();
```

```
}
```

prints a new string at memory a still of main

<stdio.h> abhiit

<conio.h> abhiit

(a) new line

{ } new { }

(b) new

; (" prints a new") string

; (" si prints last of ") string

(a) Abhis

(b) Abhiit

Program 2: Write a program to enter a character

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char a;
    clrscr();
    printf("Enter a character:");
    a = getch();
    printf("\n The character is : ");
    putchar(a);
    getch();
}
```

Program 3: Write a program to enter a string

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char [50];
    clrscr();
    printf("Enter a string:");
    gets(a);
    printf("Entered string is : ");
    putchar(a);
    getch();
}
```

OUTPUT

enter a character : a

The character is : a

56

OUTPUT

enter a string : bts

The entered string is : bts

22

OUTPUT

my name is

m

y

n

a

m

e

Program 4 : Write a program to print the string in vertical order

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    char name [10] = "my name";
    clrscr();
    printf ("my name is :");
    for (int i=0; i<10; i++)
    {
        printf ("\n");
        putchar (name [i]);
    }
    getch ();
}
```

<std.ctype> character
<std.ios> stream
() main function

: [0] : the null

: () works

(" : prints a return ") thing

: (st & . "2") new

: (st) write

: (()) dot

: (()) dot up

6
03 | 03

52

Program 5: Program to print reverse string

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char str[10];
    clrscr();
    printf ("Enter a string:");
    scanf ("%s", &str);
    rev (str);
    printf ("The reversed string is : %s", str);
    getch();
}
```

OUTPUT

Enter a string : Sakshi

The reversed string is : ihskas

PRACTICAL NO 7

59

Program 1: Write a C program to calculate area and circumference of a circle.

```
#include <stdio.h>
#include <conio.h>
void circle ( void );
void main ()
{
    clrscr();
    circle();
    getch();
}

void circle ( void )
{
    int r;
    float area ; circum;
    printf (" Enter the radius : ");
    scanf ("%d", &r);
    area = 3.14 * r * r;
    circum = 2 * 3.14 * r;
    printf ("\n Area = %.f ", area);
    printf ("\n Circumference : %.f ", circum);
    getch();
}
```

P.2
Program 2: Write a C program to find the sum of digits of entered number.

```
# include < stdio.h >
# include < conio.h >
void sum ( int n );
void main ()
{
    clrscr ();
    int n;
    printf ("Enter a number : ");
    scanf ("%d", &n);
    sum (n);
    getch ();
}

void sum ( int n )
{
    int r, s = 0;
    while ( n != 0 )
    {
        r = n % 10;
        s = s + r;
        n = n / 10;
    }
    printf ("\n sum of digits is %d", s);
    getch ();
}
```

Q8

OUTPUT

Enter two numbers: 7 8

66

Sum of two numbers: 144

Program 3:

```
#include < stdio.h>
#include < conio.h>
void sum ( int n1, int n2 );
void main ()
{
    clrscr ();
    int n1, n2;
    printf (" Enter two numbers: ");
    scanf ("%d %d", &n1, &n2);
    sum (n1, n2);
    getch ();
}

void sum ( int n1, int n2 )
{
    int a;
    a = n1 + n2;
    printf (" sum of two numbers is : %d", a);
    getch ();
}
```

Q. Program 4: Write a C program to calculate the total & average.

```
#include <stdio.h>
#include <conio.h>
void total (int m1, int m2, int m3, int m4);
void main()
{
    int a, b, c, d;
    printf ("Enter four marks: ");
    scanf ("%d%d%d%d", &a, &b, &c, &d);
    total (a, b, c, d);
    getch();
}
```

```
int a, b, c, d;
printf ("Enter four marks: ");
scanf ("%d%d%d%d", &a, &b, &c, &d);
void total (int m1, int m2, int m3, int m4);
{
```

```
    int total;
    total = m1 + m2 + m3 + m4;
    printf ("The total is %d", tot);
    average (tot);
}
```

```
float avg;
avg = tot / 4;
printf ("Average is %.f", avg);
getch();
```

}

OUTPUT
Enter
30
40
50
70
45

OUTPUT

enter four marks 20

30

40

50

The total is 140

average is 35.000

62

8

Output:

Enter value of $x = 4$

Factorial of 4 = 24

Program 5: Write a C program to find the factorial of a number

```
#include <stdio.h>
#include <conio.h>
int factorial (int n);
void main()
```

{

```
int n, fact;
printf ("Enter a number:");
scanf ("%d", &n);
fact = factorial (n);
printf ("\n Factorial of %d is %d", n, fact);
getch();
```

&

```
{ int factorial (int n)
```

```
int f;
if (n == 1)
    return 1;
else
    f = n * factorial (n-1);
return (f);
getch();
```

{

PRACTICAL NO 8

Program 1: Employee Comparison

```
#include <stdio.h>
#include <conio.h>
struct employee
{
    int eno, salary;
};

void main ()
{
    struct employee n, y;
    printf ("In Enter eno and salary :");
    scanf ("%d %d", &n.eno, &n.salary);
    printf ("In Enter eno and salary :");
    scanf ("%d %d", &y.eno, &y.salary);
    if (n.eno == y.eno & n.salary == y.salary)
        printf (" both are equal ");
    else
        printf (" both are unequal ");
    getch ();
}
```

Output:

enter rollno, name and total of student : 1832

Sakshi

100

Roll no : 1832

Name : Sakshi

Total = 100

64th

Output

Enter eno and salary = 5 200.00

Enter eno and salary = 5 200000

Both are equal

Enter eno and salary = 3 1500

Enter eno and salary = 4 25000

Both are unequal

Program 2 : fruit Structure

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

```
struct employee
```

```
{
```

```
    int eno, salary;
```

```
}
```

```
void main()
```

```
{
```

```
    struct employee n, y;
```

```
    printf("In enter eno and salary:");
```

```
    scanf("%d %d", &n.eno, &n.salary);
```

```
    printf("In enter eno and salary:");
```

```
    scanf("%d %d", &y.eno, &y.salary);
```

```
}
```

```
else
```

```
    printf("both are unequal");
```

```
getch();
```

```
{
```

Program 3: Fruit Structure

```
# include <stdio.h>
# include <conio.h>
struct fruit
{
    char name [20];
    int price, qty, total;
}
void main()
{
    struct fruit f[5];
    int k;
    clrscr();
    printf ("In enter name, price, &qty:");
    for (k=0; k<5; k++)
    {
        scanf ("%s%d%d", &f[k].name, &f[k].price, &f[k].qty);
        f[k].total = f[k].price * f[k].qty;
    }
    for (k=0; k<5; k++)
    {
        printf ("In name = %s, price = %d, qty = %d", f[k].name,
               f[k].price, f[k].qty);
    }
    getch();
}
```

Output

Enter

Apple

Mango

Banana

Cherry

Grapes

| | Name | Price | Qty |
|--|------|-------|-----|
| | 20 | 5 | |
| | 15 | 3 | |
| | 50 | 9 | |
| | 30 | 7 | |
| | 30 | 15 | |

Name = Apple , Price = 20 , Qty = 5

Name = Mango , Price = 15 , Qty = 3

Name = Banana , Price = 50 , Qty = 9

Name = Cherry , Price = 30 , Qty = 7

Name = Grapes , Price = 30 , Qty = 45

88

Output:

Roll no = 22

Name = Prakash Salary = 800

Program 5: Structures within structures

```
# include < stdio.h>
# include < conio.h>
struct employee
{
    int salary;
}
struct employee
{
    int id;
    char name [10];
    struct employee b2;
}
void main()
{
    clrscr();
    int i;
    struct employee s = {22, "prakash", {500}};
    printf ("In Rollno=%d \t Name=%s \t salary %d",
           s.id, s.name, s.b2.salary);
}
```

getch();

BJ
03/03

53

PRACTICAL NO 9

Aim: Programs on pointers in C-language

Program 1:

```
# include < stdio.h>
# include < conio.h>
void main()
{
    clrscr();
    int a=12, b=4, n, q, *p, *q;
    p=&a;
    q=&b;
    x=*p * *q - 6;
    y = 4 * (*p - *q) + 10;
    printf ("In a = %.d", a);
    printf ("In b = %.d", b);
    printf ("In x = %.d", x);
    printf ("In y = %.d", y);
    getch();
}
```

Output

$a = 12$

$b = 4$

$x = 42$

$y = 42$

88

Output

$$\text{sum} = 150$$

Output

$$x = 30$$

Program 2:

```
#include < stdio.h>
#include < conio.h>
void main()
{
    clrscr();
    int x[5] = {10, 20, 30, 40, 50};
    int *p; i; sum = 0;
    p = &x[0];
    for { i = 0; i < 5; i++ }
    {
        sum = sum + *p; p++; // pointer to next element
    }
    printf ("\n sum = %d", sum);
    getch();
}
```

program 3:

```
#include < stdio.h>
#include < conio.h>
void change (&int *p);
void main()
{
    clrscr();
    int x = 20;
    change (&x);
    printf ("\n x = %d", x);
    getch();
}
```

Program 4:

```
#include <stdio.h>
#include <conio.h>
void exchange (int* a, int* b);
void main()
{
    int x, y;
    x = 10;
    y = 20;
    printf ("In Before exchange: x=%d, y=%d", x, y);
    exchange (&x, &y);
    printf ("In After exchange x=%d, y=%d", x, y);
    getch();
}

void exchange (int* a, int* b)
{
    int t;
    t = *a;
    *a = *b;
    *b = t;
}
```

Output:

before exchange

$$x = 10, y = 20$$

70

after exchange

$$x = 20, y = 10$$

80
03/03

05

Output:

Opening the file test.c in while mode
Enter 50m text from keyboard to write
in file test.c

How are you doing?

Closing the file test.c

PRACTICAL NO 10

71

Aim: Programs on file handling

Program 1: Open file → Write and close file

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
int main()
{
    FILE fp;
    char data [50];
    printf ("Opening the file test.c in write mode");
    fp = fopen ("test.c", "w");
    if (fp == NULL)
    {
        printf (" Could not open file test.c");
        return 1;
    }
    printf ("\n Enter some text from keyboard to write in file:");
    while (stolen ( gets (data)) > 0)
    {
        fputs (data, fp);
        fputs ("\n", fp);
    }
    printf ("Closing the file test.c");
    close (fp);
    return 0;
}
```

Program 2: Using putw() & getw() functions

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

```
int main()
```

```
{
```

```
FILE *fp;
```

```
int i = 1, j = 2, k = 3, num;
```

```
fp = fopen ("test.c", "w");
```

```
putw (i, fp);
```

```
putw (j, fp);
```

```
putw (k, fp);
```

```
fclose (fp);
```

```
fp = fopen ("test.c", "r");
```

```
while (getw (fp) != EOF)
```

O/P

Name = Fresh & refresh

Age = 5

Total no of characters in file is 15

Program 3: `scanf()`, `fprintf()`, `tell()`, `rewind` functions

```
# include < stdio.h>
int main ()
{
    char name [20];
    int age, length;
    FILE * fp;
    fp = fopen ("text.txt", "w");
    fprintf (fp, "%s%d", "fresh2refresh", 5);
    length = tell (fp);
    rewind (fp);
    scanf ("%d", &age);
    scanf ("%s", &name);
    close (fp);
    printf ("Name: %s\n Age: %d\n", name, age);
    printf ("Total number of characters in file is %d", length);
    return 0;
}
```

8
03/03