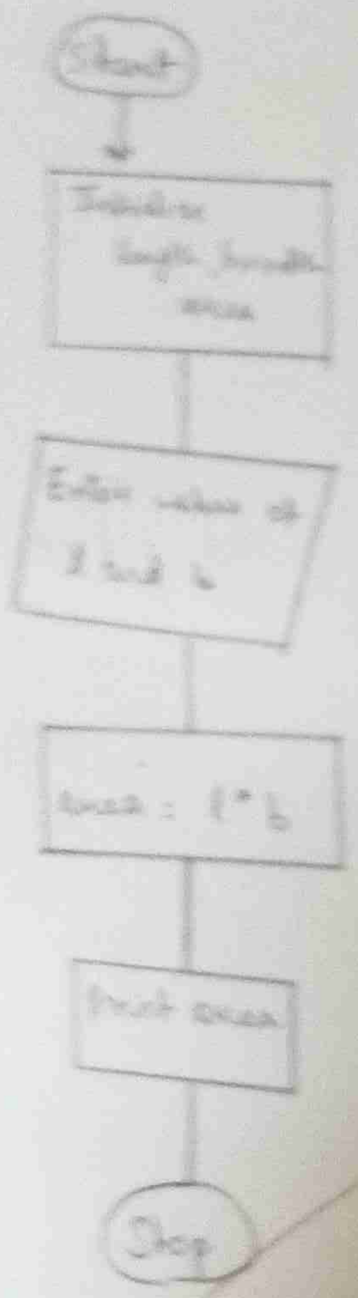
Output: Enter length and breadth 7 3
The area of rectangle 21

Flowchart:



PRACTICAL - I

AIM: Program to understand the basic datatype and input / output.

Program 1: Area of rectangle.

Algorithm:

Step 1:

Step 2: Initialize three variables length, breath, Area

Step 3: Take a input from a user and store the value int 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 ();
}
```

Program 2: Volume of sphere

Algorithm:

Step 1:

Step 2: Initialize three variable 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.

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.

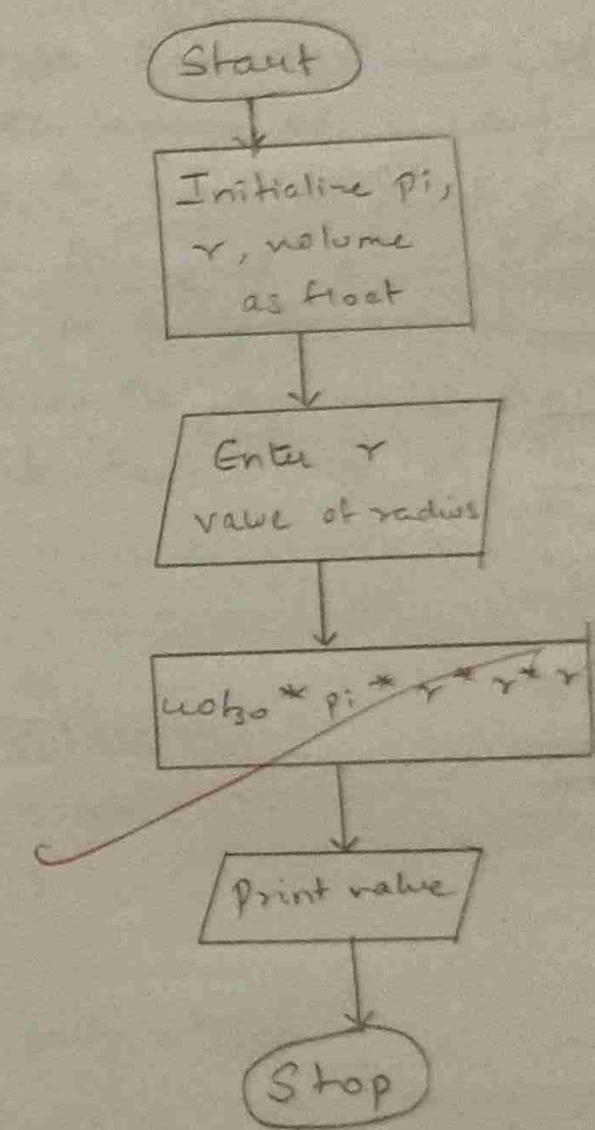
Code:

```
# include <stdio.h>
# include <conio.h>
void main()
{
  float pi, r, vol;
  pi = 3.14
  clrscr();
  printf("Enter the radius")
  scanf("%f", &r);
  vol = 4/3 * pi * r * r * r;
  printf("The volume of sphere", vol);
  getch();
}.
```


Output: Enter the radius 7

The volume of sphere: 1436.026733

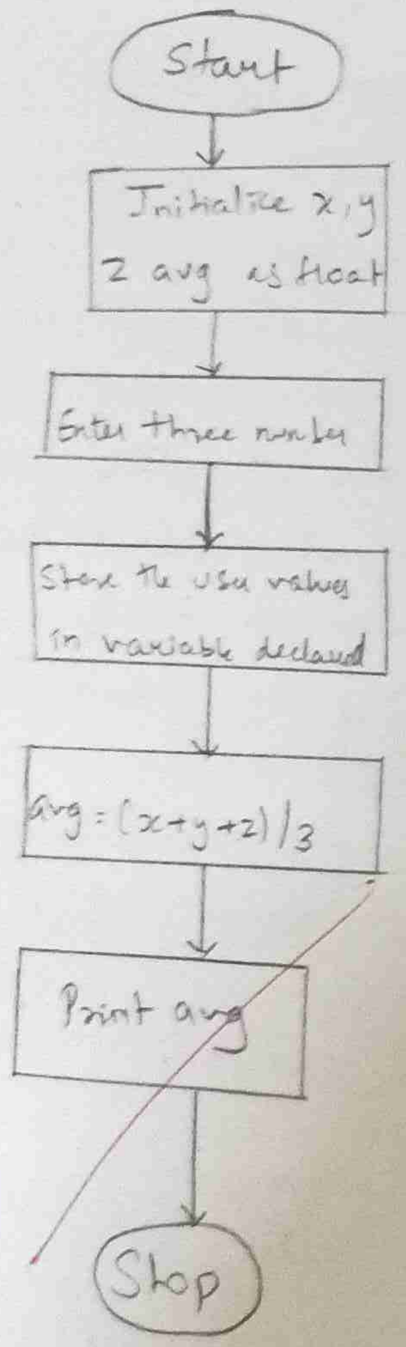
flowchart:



Output : Enter three numbers : 3 10 50

The average of three numbers : 21

flowchart :



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 avg 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, &y z);
  avg = (x+y+z)/3
  printf ("The average of three numbers :", avg);
  getch ();
}.
```

Program 4: Convert temperature from celcius to farhenheit

Algorithm:

Step 1:

Step 2: Initialize two variables for celcius and farhenheit

Step 3: Clear the screen.

Step 4: Take the input from the user in Celsius.

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

Step 6: print the fahrenheit value

Step 7: End.

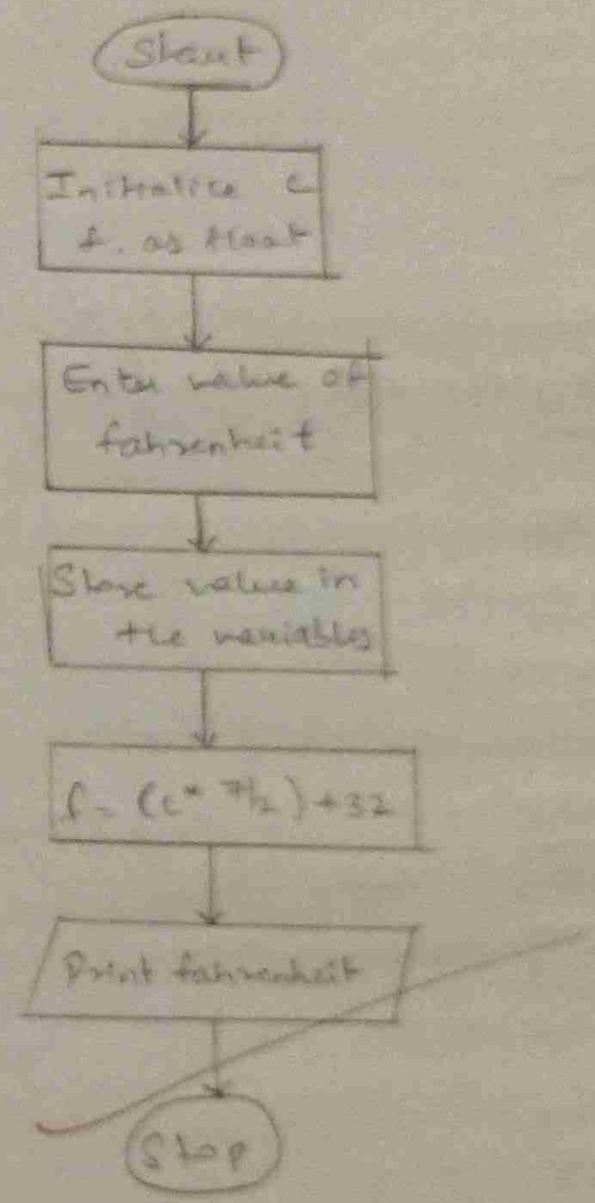
Code:

```
# 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 fahrenheit is", f);
 getch ();
}
```


Output: Enter the value of celsius 31

The fahrenheit is: 87.8

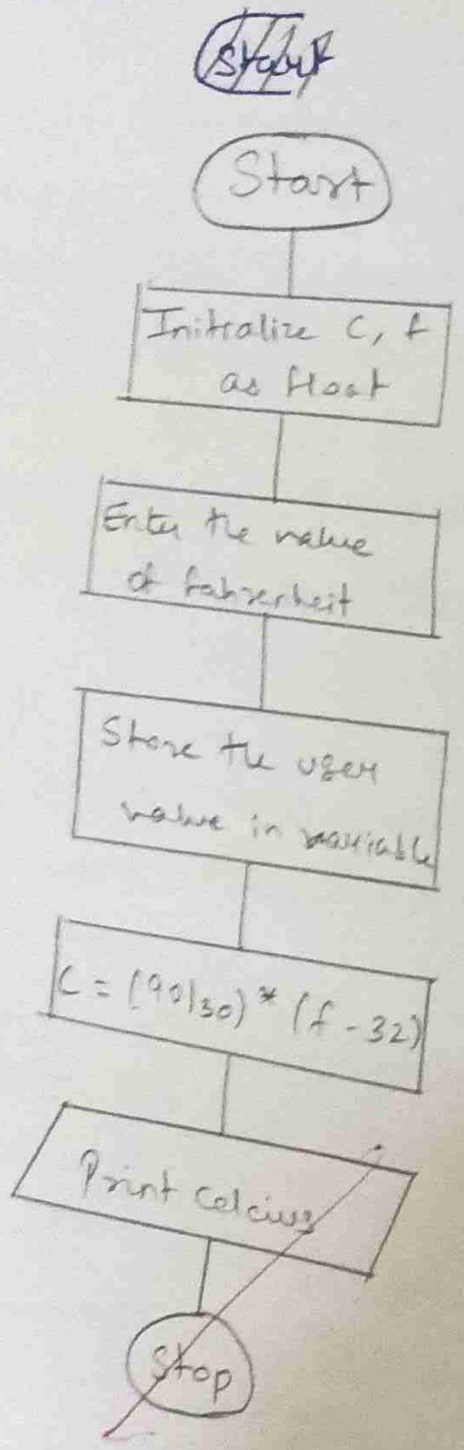
Flowchart:



Output: Enter the value of fahrenheit 80.

The celsius is: 26.666667.

flowchart:



Program 5: Convert the temperature from fahrenheit to celsius.

Algorithm:

Step 1:

Step 2: Initialize two variables with suitable datatype for celsius and fahrenheit.

Step 3: Clear the screen.

Step 4: Take input from the user for fahrenheit.

Step 5: Store the value of fahrenheit which is initialized at step 2.

Step 6: Perform the celcius equation 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 fahrenheit");
  scanf ("%f", &f) ;
  c = (9.0/5.0) * (f - 32);
  printf ("The celsius is:", c);
  getch ();
}
```

21/01

PRACTICAL -2

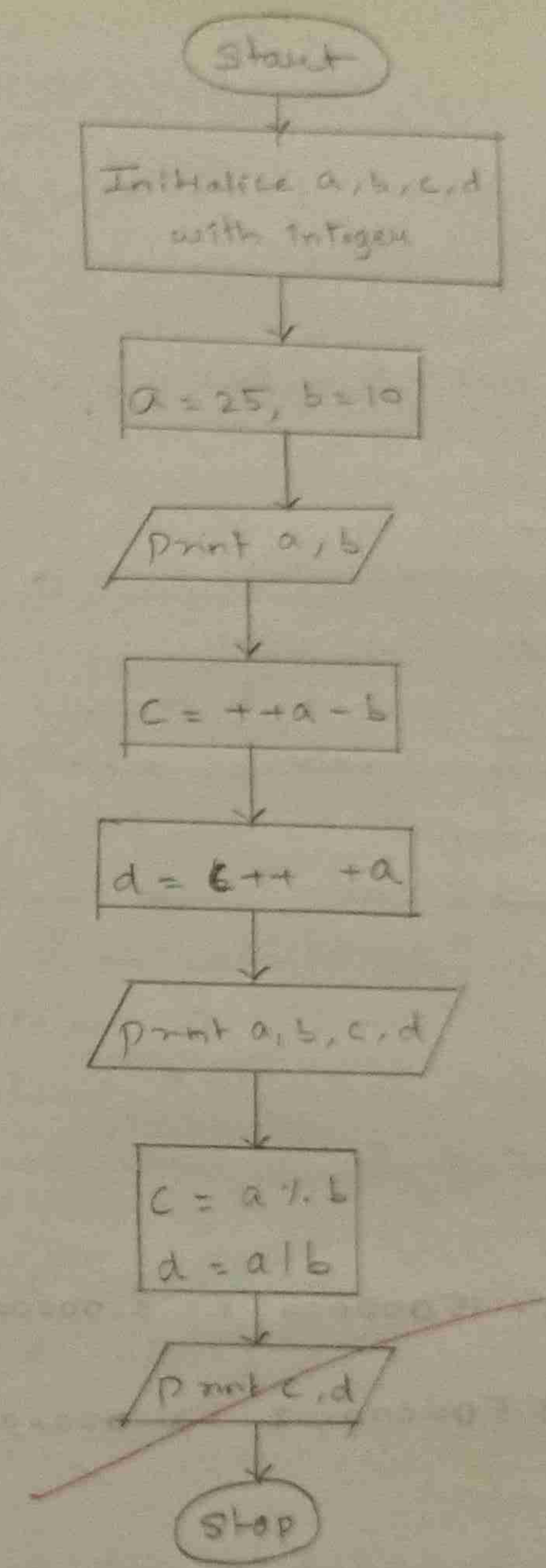
AIM: Programs on operation and expressions

Program - 1 :

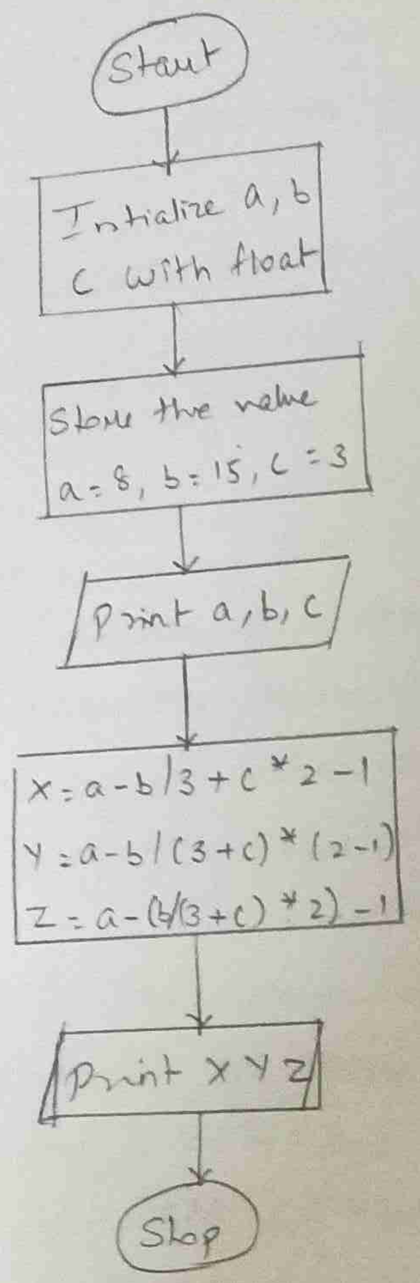
Algorithm:

Step 1: Initialize four variables using datatype int.
Step 2: clear the screen.
Step 3: Store the value 25 in a and 10 in b.
Step 4: Print 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
```



output: a=25, b=10
a=26, b=11, c=16, d=36
c=4, d=2



Output: a = 8.000000 , b = 15.000000 c = 3.000000

x = 8.000000 , y = 5.500000 , z = 2.000000


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

Program 2:

Algorithm:

Step 1: Initialize variables a, b, c with value a = 8 b = 15 , 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 value of x, y, z.

```
# include <stdio.h>
# include <conio.h>
void main ()
{
 float a, b, c;
 a = 8;
 b = 15;
 c = 3;
 printf (" \n 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;
```

```
printf("\n x= %f , y = %f , z = %f", x, y, z);
getch();
}
```

Program 3

Algorithm:

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

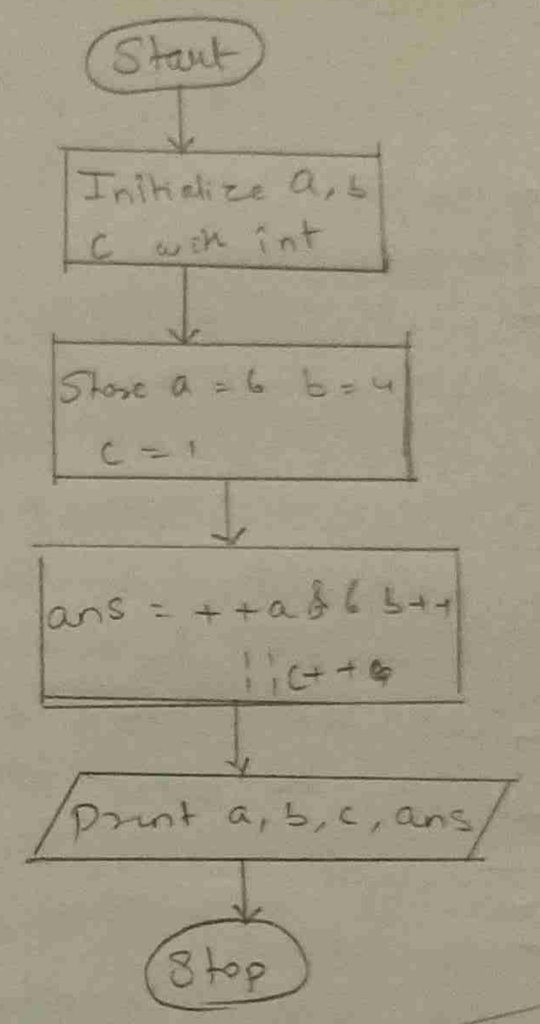
Step 2 : Clear the screen.

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

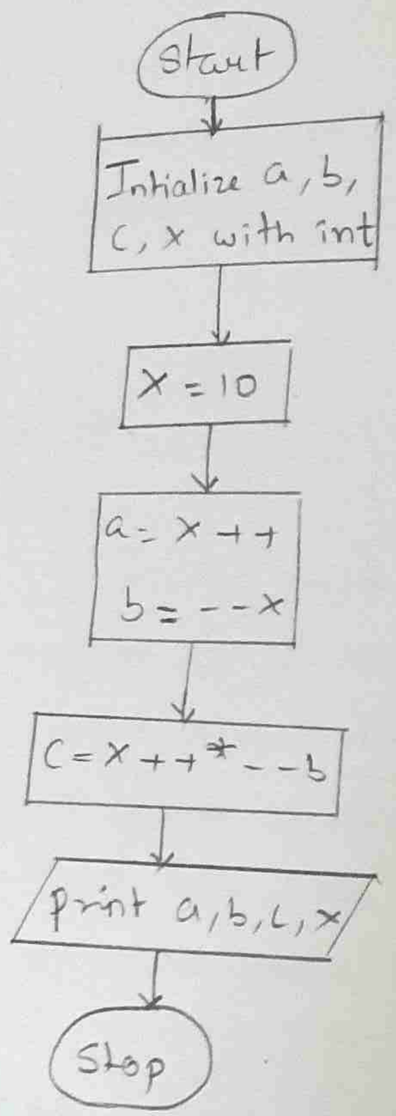
Step 4 : program expression ++a && b b++ || c++; and store the value in ans.

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 b++ || c++;
printf("a = %d , b = %d c = %d, ans = %d ; a, b, c, ans);
getch();
}.
```

Output : a = 7, b = 5, c = 1, ans = 1



Output : a = 10 , b = 9 , c = 90 , x = 11 .

Program 4:

Algorithm :-

Step 1 : Intialize variables a, b, c, x 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.

Step 5 : pre decrement the value of x and store in b.

Step 6 : Perform x++ * --b and store in c.

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, x);
  getch ();
}
```

21/01

PRACTICAL - 3.

AIM - Program on decision making & branching.

Program 1: Check whether number is odd or even.

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

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

Output:

Enter the alphabet: i

i is a vowel

Enter the alphabet: 3

3 is a consonent.

Output : Enter 3 nos. 2
7
1
b is greater.

Program 4:

```
# include <stdio.h>
# include <conio.h>
void main ()
{
 int a,b,c ;
 clrscr ();
 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));
 printf ("\n b is greater";
 else
 printf ("\n c is greater");
 getch();
}.
```

Program 5: Program to enter single digit decimal number of 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");
```

Output

Enter single digit no: 1

one

Enter single digit no: 15

error.


```
else if (n==9)
printf ("\n nine");
else
printf ("\n error");
getch();
}
```

Program 7: Program to perform addition substraction multiplication using switch case.

```
# include <stdio.h>
# include <conio.h>
void main ()
{
clrscr ();
int a, b, choice ;
printf ("\n select your choice");
printf ("\n 1. addition");
printf ("\n 2. Substraction");
printf ("\n 3. Multiplication");
printf ("\n 4. Division");
printf ("\n 5. Exit");
Scanf ("%d", choice);
if (choice > -1 && choice < -4)
{
printf ("\n Enter value of a & b");
Scanf ("%d %d", &a, &b);
}
Switch (choice)
{
case 1:
 r = a+b ;
printf ("\n %d + %d = %d", a,b,r);
break;
Case 2:
r = a-b;
```

Output:

Enter your choice

2

Enter value of a & b 8

10

-2


```
printf ("\n %d - %d = %d", a,b,r);
break;
Case 3;
r = a*b;
printf ("\n %d * %d = %d", a, b, r);
break;
case 4:
r = a/b
printf ("\n %d / %d = %d", a,b,r);
break;
default:
printf ("\n No operation"),
break;
}
getch();
}
```

04/02

Practical - 4

Aim:- 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();
}.
```

Output :-

2
4
6
8
10
12
14
16
18
20

Output:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```


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);
++k;
}
printf ("\n");
i++;
}
getch();
}
```

Program 3.

Code :-

```
# 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
{
 x = i%2;
 if (x==1)
{
 sum = sum + i;
}
++i;
}
while (i<=n)
printf("The sum of all odd no are "%d", sum);
getch();
}
```


Output

Enter the value of n : 10

The sum of all odd no are 25

Output:-

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


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 ();
 }
```

Program 5.

```
# include <stdio.h>
# include <conio.h>
void main ()
{
int a,b, t, i ;
clrscr();
a = 1;
b = 0;
for (i = 3; i <= 20 ; i++)
{
t = a+b;
printf("\n %d",t);
a = b;
b = t ;
getch ();
}
```

Output:

```
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
```

04/02

Output.

Enter 10 data of array:

1 2 3 4 5 6 7 8 9 10 11

largest = 11.

Practical : 5

AIM - To understand the concept of Arrays.

→ ONE DIMENSIONAL ARRAYS :-

Program 1 : Find the largest number in an array of 10 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();
```

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("\n 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("\n Smallest : %d", s);
getch();
}
```

Output

12 56 90 7 11 46 77 60 10 34

Smallest : 7

Output

Enter 10 data of array.

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

No. of positive no = 6.

Program 3: Find number of positive numbers present in an 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();
}
```

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

```
#include <stdio.h>
#include <conio.h>
void main ()
{
clrscr();
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 negative no = %d", c);
getch();
}
```

Output

Enter 10 data of array:

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

No. of negative = 5

Output:

Enter 10 data of array:

1 2 3 4 5 6 7 8 9 10

No. of odd no: 5

Program 5:

Write a program to find numbers of odd no. present in array of 10 data :-

```
# Include <stdio.h>
# include <conio.h>
void main ()
{
int i, r, c , a[10];
clrscr ();
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++)
{
r= a [i] %2;
if (r ==1)
c++;
}
printf (" \n No. of odd no: %d", c);
getch ();
}.
```

Program 6: Find no. of even numbers.

```
#include <stdio.h>
#include <conio.h>
void main ()
{
int i, r, c, a[10];
clrscr();
printf ("\n Enter 10 data of arrays");
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 ("\n No. of even no= %d", c);
getch();
}.
```


Output:

Enter 10 data of array:

1 2 3 4 5 6 7 8 9 10

No. of even no: 5

Output :

Enter 10 data of array:

1 2 3 4 5 6 7 8 9 10

Sum = 55

AVERAGE = 5.500000.

Program 7.

Write a program to find average of 10 data of array.

```
# include <stdio.h>
# include <conio.h>
void main()
{
clrscr();
int a[10], i, s;
float avg;
printf(" \n 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 = %f ", avg);
 getch();
}
```

03/03

Practical No : 6

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 ("\n Enter your name");
scanf ("%s", &name);
printf ("\n My name is : %s", name);
getch ();
}.
```


Output :

Enter your name : Nisha.

My name is : Nisha.

Output :
Enter a character : a
The character is : a

Output :
Enter a string : bts
The entered string is : bts.

Program 2 : Write a programe to print the entered character.

```
# include <stdio.h>
# include <conio.h>
void main ()
{
char a ;
clrscr ();
printf (" Enter a character:");
a = getchar ();
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:");
get (a).
printf (" \n Entered string is:");
put (a) ;
getch();
```

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();
}.
```

Output:

my name is

m
y
n
a
m
e

Output:

Enter a string : Nisha

The reversed string is : ahsiN

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);
strev (str);
printf ("The reversed string is : %s", str);
getch();
}
```

03/03

Practical : 7.

AIM : Programs using USER - defined function.

Program 1: Area & circumference of a circle.

```
# include <stdio.h>
# include <conio.h>
void circle (void);
void main ()
{
clrscr();
circle ();
getch();
}
void circle (void)
{
int r;
float a,c;
printf ("\n Enter value of r:");
scanf ("%d", &r);
a = 3.14 * r * r;
c = 2 * 3.14 * r;
printf(" Area = %f", a);
printf(" Circumference = %f", c);
}
```

Output :

Enter value of r = 5

Area = 78.500000.

Circumference = 31.400000.

Output:

Enter num = 5

Entered num = 5

Program 2: Print digits of the entered numbers.

```
# include <stdio.h>
# include < conio.h>
int get_no (void);
void main ()
{
clrscr ();
int m;
m = get_no ();
printf (" \n Entered_num = %d", m);
getch();
}
int get_no (void)
{
int num;
printf (" Enter num :");
scanf ( " %d" , & , num);
return ( num);
}
```

13.

Program 3 : Sum of digits of entered numbered.

```
# include <stdio.h>
# include <conio.h>
 void abc (int n);
void main ()
{
 clrscr ();
int n;
printf (" Enter number :");
scanf (" %d", &n);
abc (n);
getch ();
}
void abc (int n)
{
int r, s = 0
while (n != 0)
{
r = n % 10;
s = s+r
n = n/10
}
printf (" Sum of digits = %d", s);
}.
```

Output:

Enter number = 31

Sum of digits = 4.

Output:

Enter Value of x, y, z : 4 6 9

Average : 6.333333

Program 4: Average of 3 (entered numbers)

```
# include <stdio.h>
# include <conio.h>
void average (int sum);
void sum (int a, int b, int c);
void main ()
{
clrscr ();
int x, y, z;
printf("\n Enter value of x,y,z);
scanf(" %d %d %d", &x, &y, &z);
sum (x, y, z);
getch ();
}
void sum ( int a , int b , int c );
{
int s;
s = a+b+c;
average (s);
}
void average (int sum)
{
float avg;
avg = sum/3.0;
printf(" \n Average : %f", avg);
} }
```

Program 5: Factorial of number using RECURSION

```
#Include <stdio.h>
# Include <conio.h>
int factorial (int n);
void main ()
{
clrscr();
int x, fact;
printf ("In Enter value of x:");
scanf ("%d", &x);
fact= factorial (x);
printf ("In Factorial of %d = %d", x, fact);
getch();
}
int factorial (int n);
{
int f;
if (n==1)
Return (1)
else
f = n* factorial (n-1)
Return (f)
}.
```


Output:

Enter value of x = 4

Factorial of 4 = 24.

03/03

output:

Enter rollno, name and total of student : 1759

Payal
100

Rollno = 1759
Name = Payal
Total = 100.

AIM : Programs on structures.

Program 1 : Student Structures

```
# include <stdio.h>
# include <conio.h>
struct student
{
int roll no ;
char name [20];
int total ;
};
void main ()
{
struct student *x;
clrscr();
printf( " \n Enter name ,rollno & total of student : ");
printf( " %d %s %d " , & x.rollno ,& x.name, & x.total
printf ( " Rollno = %d " , x . roll no);
printf ( " \n Name = %s " , x . name);
Printf ( " \n Total = %d", x . total);
getch ();
}
```

Program 2: Employee comparison.

```
# include <stdio.h>
# include <conio.h>
struct employee
{
int eno , salary;
};
void main ()
{
struct employee n, y;
printf ("\n Enter eno and salary:");
scanf(" %d %d", & n.eno , & n. salary);
printf(" \n 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");
}
```

Output

Enter eno and salary = 5 20000

Enter eno and salary = 5 200000

Both are equal.

Enter eno and salary = 3 150000

Enter eno and salary = 4 25000

both are unequal.

Output:

Enter Name Price & qty

Apple	20	5
Mango	15	3
Banana	50	9
cherry	30	7
Grapes	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 = 15

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 ("\n 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 ("\n name = %s ; price = %d ; qty = %d", f[k] name,
        f[k].price, f[k] qty);
}
getch();
}
```

Program 4: Cricketers & their names

```
# include <stdio.h>
# include <conio.h>
# include <string.h>
struct cricket
{
char pname [20] , tname [20];
int average ;
};
void main ()
{
clrscr();
struct cricket p[5],t ;
int i, k , x;
printf ("\n Enter records of 5 players:");
for (i=0 ; i<4 ; i++)
scanf ("\n %s %s %d \n", & p[i].name ; & tname , & p[i].avg);
for (i=0 ; i<4 ; i++),
{ for ( k=i+1 ; k<5 ; k++)
  { x = strcmp (p[i].tname , p[k].tname);
    if (x>0)
    { t = p[i];
      p [i] = p[k];
      p[k] = t ; }}}
printf ("\n teamwise player names \n");
for (i=0 ; i<5 ; i++)
{ printf ("\n %s %s %d \n", p[i].pname , p[i] tname , avg);
}
getch ();
}
```

Output:

Enter records of 5 players:-

Ms Dhoni	India	100
Virat	India	100
Rohit	India	100
Shikhar	India	100
Rahane	India	100

Teamwise	Player	Name
Ms Dhoni	India	100
Virat	India	100
Rohit	India	100
Shikhar	India	100
Rahane	India	100

Output:

Rollno = 22 Name = Prakash salary = 800

Program 5. Structure within structure

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

03/03

Practical - 9.

AIM - Programs on pointers in C- language.

Program 1 :

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

Output

a = 12

b = 4

x = 42

y = 42

Output:
sum = 150

Output:
x = 30

Program 2 :-

```
# ~~inclu~~ 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 = p + 1;}
printf ("\n sum = %d", sum);
getch();
}
```

Program 3: Pointers as function argument.

```
# include <stdio.h>
# include <conio.h>
void change (int * p)
void main()
{
clrscr();
int x = 20;
change ( & x);
Printf ("\n x = %d", x);
getch(); }
Void main (int * p)
{ * p = * p + 10; }
```

Program 4:

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

After exchange x = 20, y = 10

03/03

Output:

Opening the file test.c in write mode

Enter some text from keyboard to write in file test.c

Hi, How are you doing?

Closing the file txt.c

03/23

PRATICAL NO: 10

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 (" In Enter some text keyboard to write in file :");
while ( strlen ( gets ( data )) > 0)
{ fputs (data, fp);
fputs ("\n", fp); }
printf (" Closing the file test .c");
fclose (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)
```

15.

Output:

Name: Fresh 2 refresh

Age : 5

Total no. of characters in file is 15.

Program 3: fscanf (), fprintf (), ftell (), rewind functions.

```
# include <stdio.h>
int main ()
{
char name [20];
int age, length;
FILE * fp;
fp = fopen ("text.txt", "w");
fprintf (fp, "%s %d", "fresh 2 refresh", 5);
length = ftell (fp);
rewind (fp);
fscanf (fp, "%d", &age);
fscanf (fp, "%s", &name);
fclose (fp)
printf (Name : %s \n Age %d \n", name, age);
printf ("Total number of character in file is %d", length
return();
}
```

03/03