Program 1: To read radius of circle and to find the area of circumference.

Algorithm:

Step 1: Start.

Step 2: Initialize the Variable Ci, Pi=3.142,Area.

Step 3: Read the radius value.

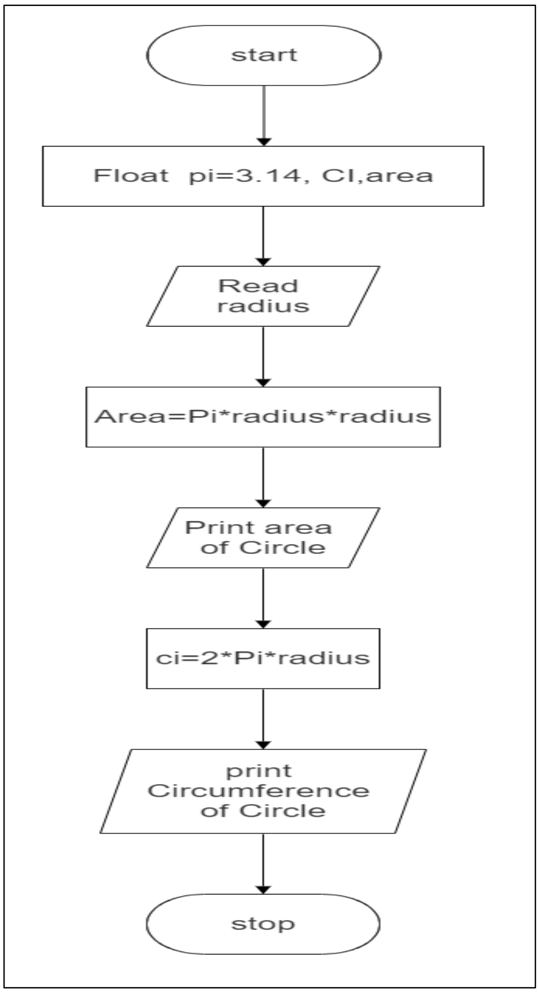
Step 4: Calculate area=PI*radius*radius.

Step 5: Print the area of the Circle.

Step 6: Calculate the Circumference of Circle Ci=2*pi*radius.

Step7: Print the Circumference of Circle.

Step 8: End



```
#include<stdio.h>
#include<conio.h>
void main(){
int radius;
float area, ci, pi=3.142;
clrscr();
printf("enter a value for radius: \n");
scanf("%d", &radius);
area=pi*radius*radius;
printf("Area of the crircle is :%f \n", area);
ci=2*pi*radius;
printf("The circumference of a circle is :%f \n",ci);
getch();
}
```

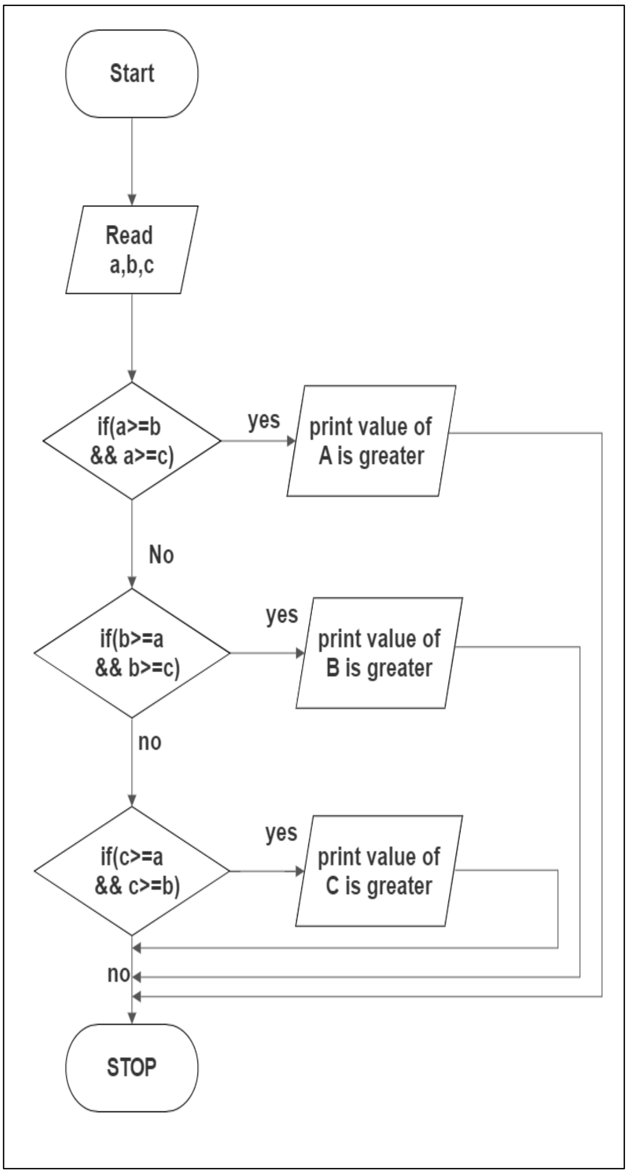
Output:

```
enter a value for radius: 2
Area of the crircle is :12.568000
The circumference of a circle is :12.568000
```

Program 2: To read the numbers and find the biggest of three.

Algorithm:

```
Step 1:Start
Step 2: Initialize the Variable a,b,c.
Step 3: Read numbers  a, b, &c.
Step 4: Check if A is greater than B and C,
        If true Print A is greater else go to step 5.
Step 5: Check if  B is greater than A and C.
        If true print B is greater else go to step 6.
Step 6: Check if C is greater than A and B.
        Print C is greater and go to step 7.
Step 7: End
```



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

int a,b,c;
clrscr();
printf("enter three numbers:");
scanf("%d %d %d", &a,&b,&c);
if(a>=b && a>=c)
printf(" %d is greater \n",a);
if(b>=a && b>=c)
printf(" %d is greater \n",b);
if(c>=a && c>=b)
printf("%d is greater\n",c);
getch();
}
```

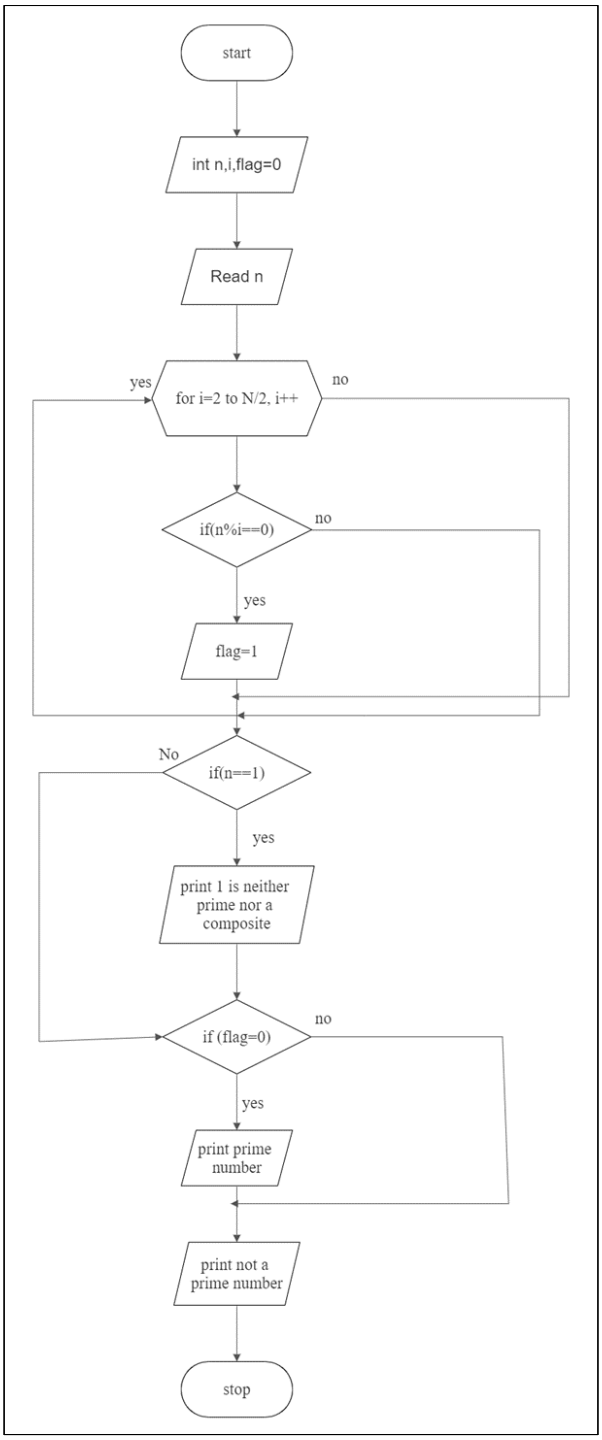
Output:

```
  enter three numbers: 120 150 130
 150 is greater
  enter three numbers: 150 100 110
 150 is greater
```

Program 3: To check whether the given number is prime or not.

Algorithm:

```
Step 1: Start.
Step 2:Intialize the number n,i,flag=0.
Step 3:Read the number n.
Step 4:For i=2 to i<n/2 go to step 5.
Step 5:if n%i=0 then set flag=1 and Exit. End loop.
Step 6:if n==1 then
            Print 1 is neither nor a Composite.
Step 7:Else if(flag==0) then
           Print that given number is Prime.
           Else Print  its not a Prime number.
Step 8:End.
```



```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,i,flag=0;
clrscr();
printf("Enter a number:");
scanf("%d",&n);
for(i=2;i<=n/2;i++)
{
if(n%i==0)
{
```

```
flag=1;
break;
}
}
if(n==1)
{
printf("1 is neither prime nor a composite");
}
else
{
if(flag==0)
printf("%d is a prime number.",n);
else
printf("%d is not a prime number.",n);
}
getch();
}
```

Output:

```
 Enter a number:13
13 is a prime number.
Enter a number:1
1 is neither prime nor a composite
Enter a number:4
4 is not a prime number.
```

Program 4: To read a number, find the sum of the digits, reverse the number and check it for palindrome.

Algorithm:

Step 1: Start.

Step 2:Initialize the variable n,sum=0,reverse=0,temp,rem.

Step 3:Read number n.

Step 4: Assign the value to Variable temp=n.

Step 5: Repeat step 5 until n!=0.

Step 6: Set rem=n%10.

Set reverse=reverse*10+rem.

Set n=n/10.

Set sum=sum+rem.

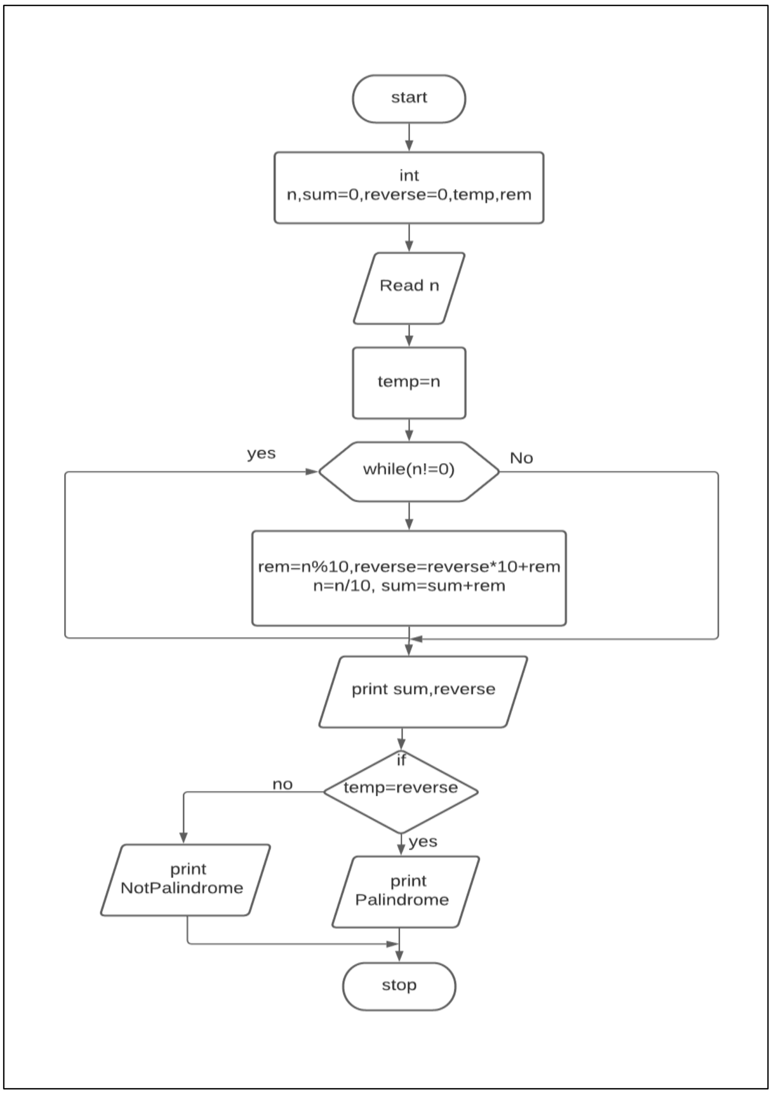
Step 7: Print the sum, Reverse value

Step 8: if temp=reverse ,print the given number is a Palindrome.

If not Print the given number is not a Palindrome.

Step 9: End

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,sum=0,reverse=0,temp,rem;
clrscr();
printf("Enter a positive integer:");
scanf("%d", &n);
temp=n;
while(n!=0)
{
rem=n%10;
reverse=reverse*10+rem;
n=n/10;
sum=sum+rem;
```



```
}
printf("sum=%d \nReverse=%d \n", sum,reverse);
if(temp==reverse)
printf("The given number is a pallindrome %d", temp);
else
printf("The given number is not a pallindrome %d",temp);
getch();
}
```

Output:

```
Enter a positive integer:123
sum=6
Reverse=321
The given number is not a pallindrome 123

Enter a positive integer:121
sum=4
Reverse=121
The given number is a pallindrome 121
```

5) To read numbers from keyboard continuously till the users presses 999 and to find the sum of only positive numbers.

Algorithm:

Step1: Initialize the variable num, sum=0.

Step2: Repeat the Step 3to 5,Read the num value.

Step3:If(num>0 && num!=999) then Print sum=sum + num.

Step 4:Print the sum value.

Step 5: While(num!=999) [End of step2 loop].

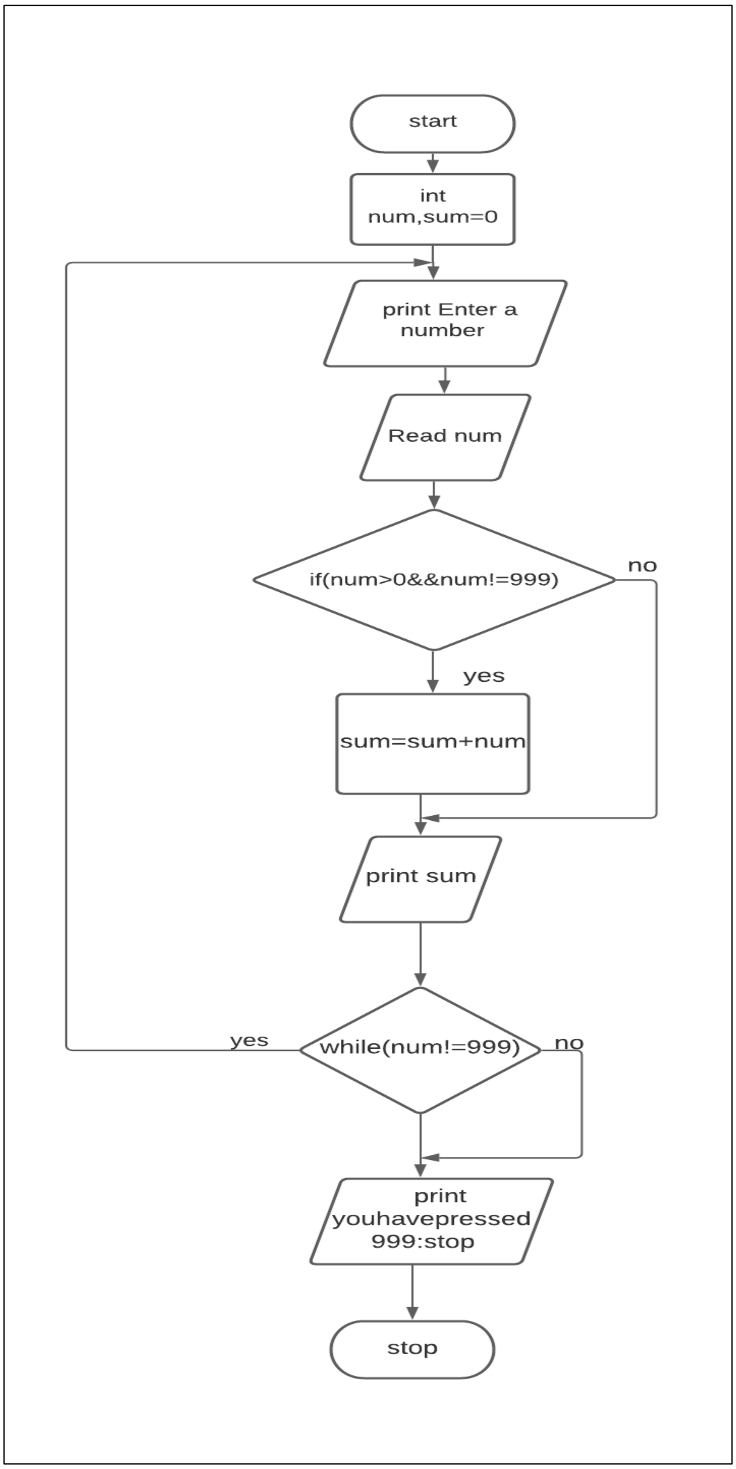
Step 6: Print the message “you have Pressed 999”.

Step 7:Stop.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int num,sum=0;
clrscr();
do
{
printf("enter a number:");
scanf("%d",&num);
if(num>0 && num!=999)
sum=sum+num;
printf("sum=%d\n",sum);
}
while(num!=999);
printf("You have pressed 999:stop");
getch();
}
```

Output:

```
enter a number:20
sum=20
enter a number:30
sum=50
enter a number:-10
sum=50
enter a number:999
sum=50
You have pressed 999:stop
```

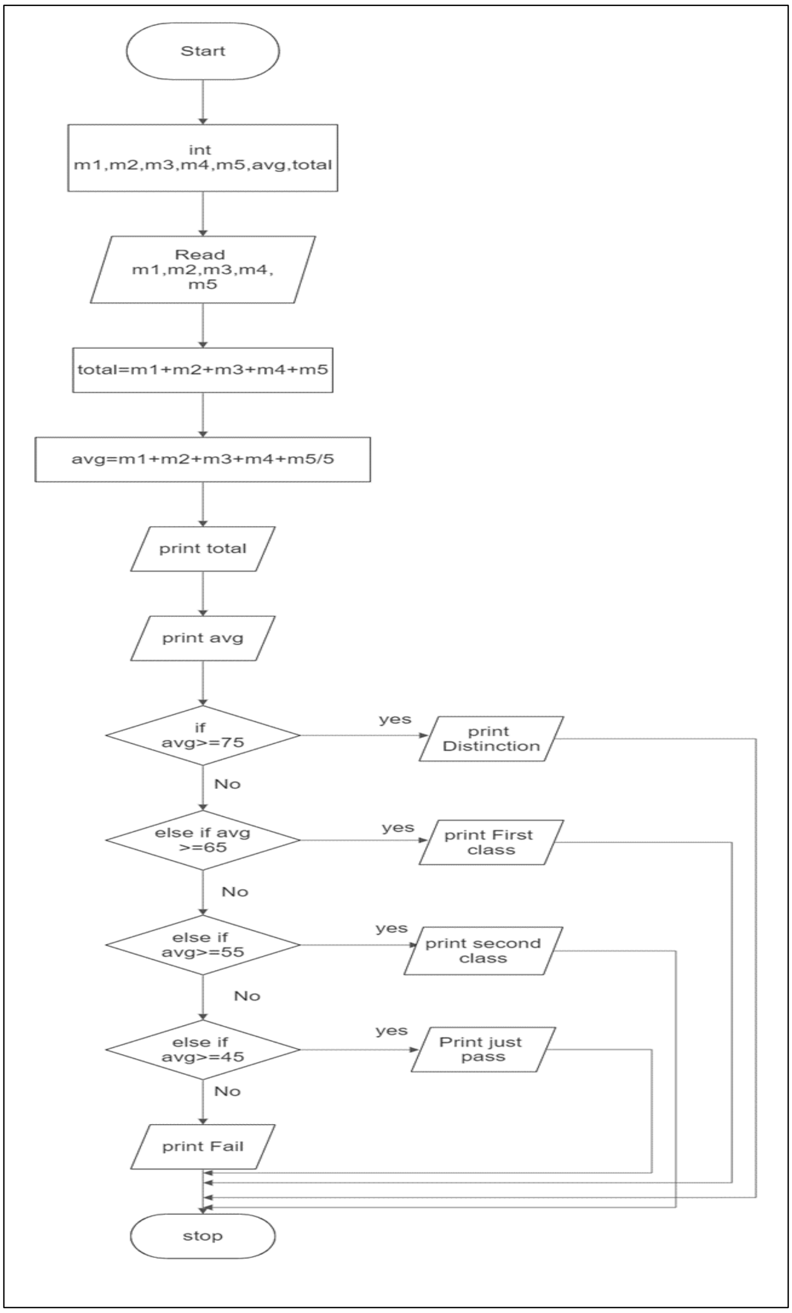


Program 6: To read percentage of marks and to display appropriate message using nested else-if ladder.

Algorithm:

```
Step 1: Start the program.
Step 2: Declare the integer variable,m1,m2,m3,m4,m5,avg,total.
Step 3:Read the m2,m3,m4,m5 marks.
Step 4: Add the marks total=m1+m2+m3+m4+m5.
Step 5: Calculate the average, avg=m1+m2+m3+m4+m5/5.
Step 6: Print the total marks.
Step 7: Print the average marks.
Step 8:If average marks>=75 print Distinction.
Step 9:If average marks>=65 print the First class.
Step10:If the average marks>=55 print the Second class.
Step11:If the average marks>=45 print Pass.
Step12:If the average marks< 45 print Fail.
```

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

void main()
{
int m1,m2,m3,m4,m5, avg,total;
clrscr();
printf("Enter five subjects marks: ");
scanf("%d%d%d%d%d", &m1, &m2, &m3, &m4, &m5);

total =m1+m2+m3+m4+m5;
avg=(m1+m2+m3+m4+m5)/5;
printf("Total Marks: %d \n",total);
printf("Average: %d \n",avg);

if(avg>= 75)
{
printf("Distinction");
   }
   else if(avg>=65)
   {
printf("First class");
   }
   else if(avg>= 55)
   {
printf("Second class");
   }
else if(avg>= 45)
   {
printf("Just pass");
   }
   else
   {
printf("Fail");
   }
getch();

}
```



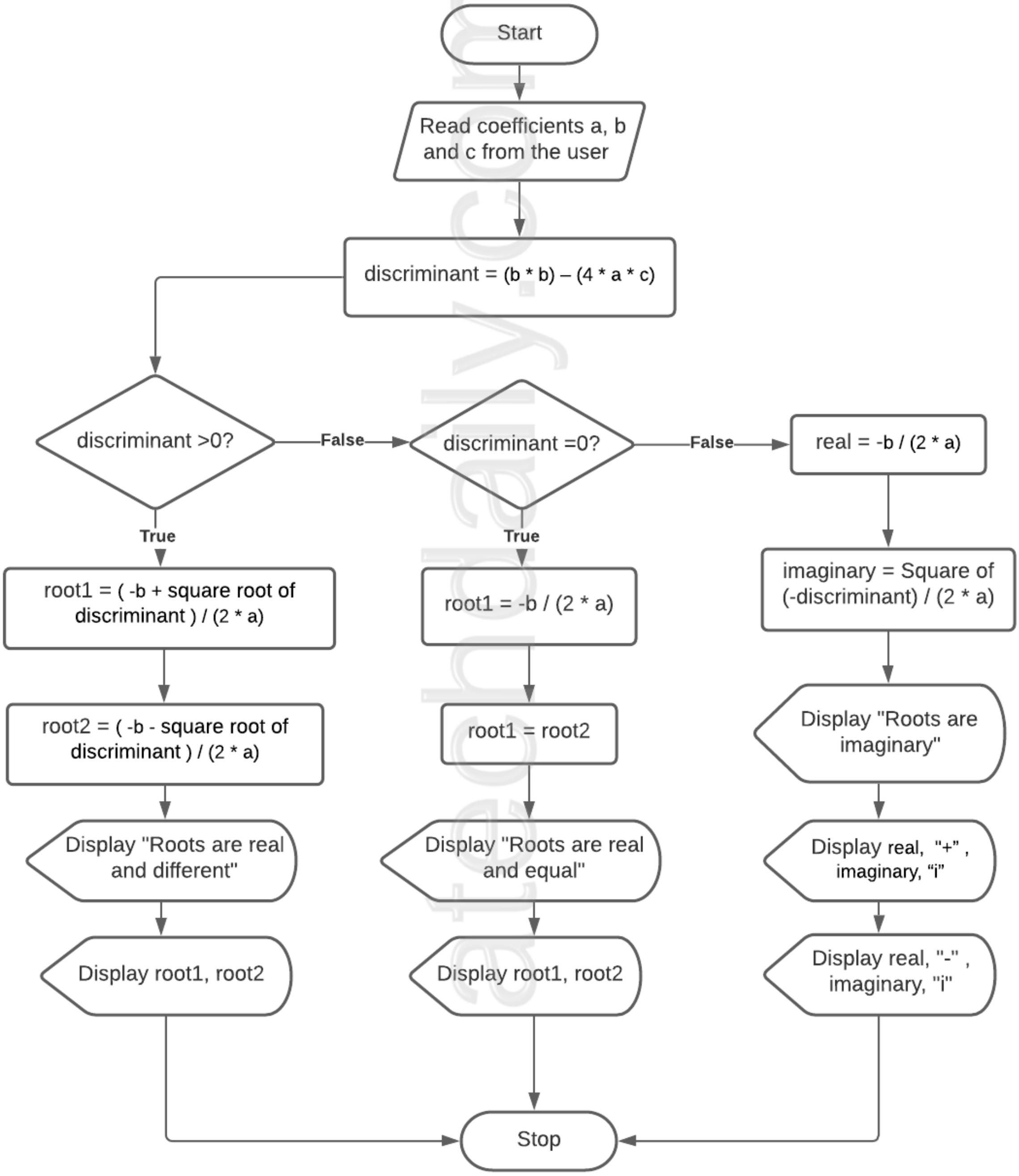
Output:

```
Enter five subjects marks: 70 80 90 95 85
Total Marks: 420
          Average: 84
Distinction
Enter five subjects marks: 10 20 30 40 10
Total Marks: 110
Average: 22
Fail
```

Program 7: Program to find the quadratic equation

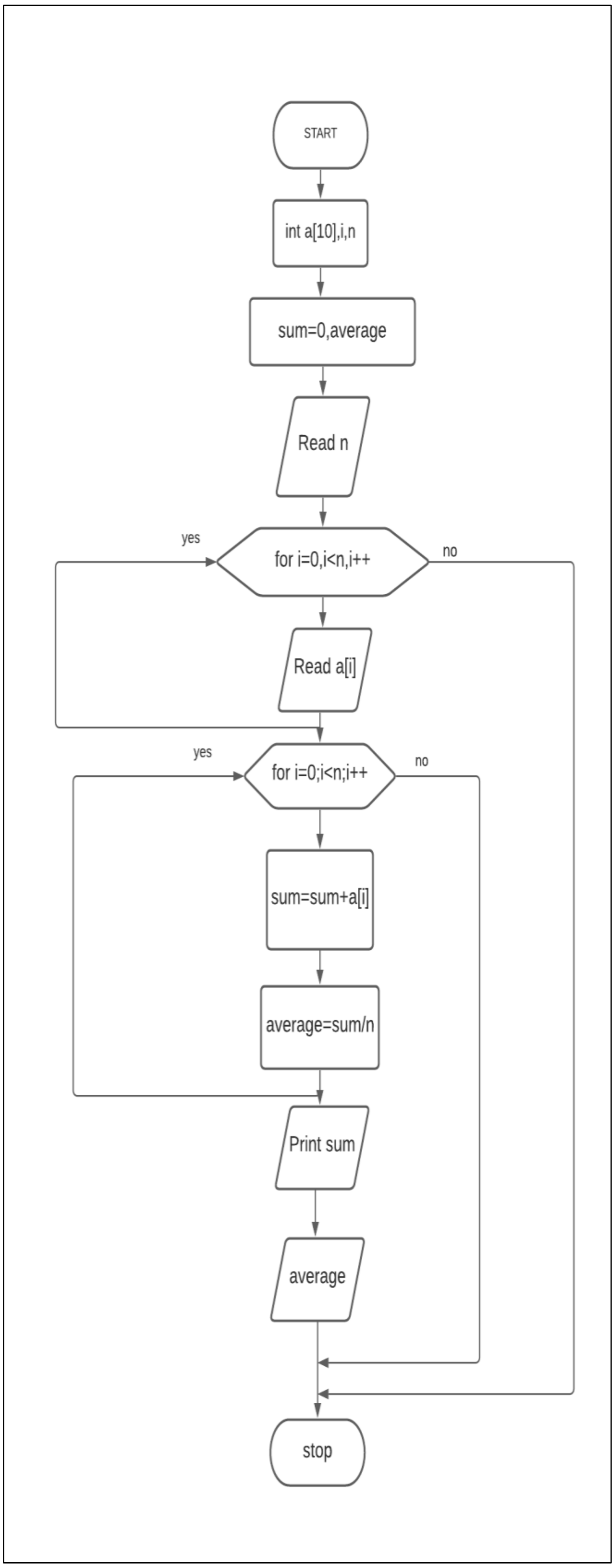
Algorithm:

```
Step 1: Start
Step 2: Read the coefficients of the equation, a, b and c from the user.
Step 3: Calculate discriminant = (b * b) – (4 * a * c)
Step 4: If discriminant > 0:
        Calculate root1 = ( -b + sqrt(discriminant)) / (2 * a)
        Calculate root2 = ( -b - sqrt(discriminant)) / (2 * a)
        Display “Roots are real and different”
        Display root1 and root2
Step 5: Else if discriminant = 0:
        Calculate root1 = -b / (2 *a)
        root2 = root1
        Display “Root are real and equal”
        Display root1 and root2
Step 6: Else
        6.1: Calculate real = -b / (2 * a)
        6.2:Calculate imaginary = sqrt(-discriminant) / (2 * a)
        6.3: Display “Roots are imaginary”
        6.4: Display real, “±” , imaginary, “i”
Step 7: Stop
```


Start
Read coefficients a, b and c from the user
discriminant = (b * b) – (4 * a * c)
discriminant >0?
False
discriminant =0?
False
real = -b / (2 * a)
True
True
root1 = (-b + square root of discriminant) / (2 * a)
root1 = -b / (2 * a)
imaginary = Square of (-discriminant) / (2 * a)
root2 = (-b - square root of discriminant) / (2 * a)
root1 = root2
Display "Roots are imaginary"
Display "Roots are real and different"
Display "Roots are real and equal"
Display real, "+" , imaginary, "i"
Display root1, root2
Display root1, root2
Display real, "-" , imaginary, "i"
Stop

```
#include <stdio.h>
#include<conio.h>
#include <math.h>
void main()
{
    float a, b, c, discriminant, root1, root2, realPart, imaginaryPart;
    clrscr();
   printf("Enter coefficients a, b and c: ");
   scanf("%.2f %.2f %.2f",&a, &b, &c);
   discriminant = b*b-4*a*c;
    if (discriminant > 0)
    {
        root1 = (-b+sqrt(discriminant))/(2*a);
        root2 = (-b-sqrt(discriminant))/(2*a);
            printf("Roots are real and different");
printf("root1 = %.2f and root2 = %.2f",root1 , root2);
    }
    else if (discriminant == 0)
    {

root1 = root2 = -b/(2*a);
printf("Both the roots are real and equal\n");
printf("root1 = root2 = %.2f;", root1);
    }
    else
    {
realPart = -b/(2*a);
imaginaryPart = sqrt(-discriminant)/(2*a);
printf("root1 = %.2f+%.2fi and root2 = %.2f-%.2fi", realPart, imaginaryPart, realPart, imaginaryPart);
    }
getch();
}
```

Program 8: To read marks by n students and find the average of marks using single dimensional array.



Algorithm:

Step 1: Start.

Step 2: initialize the array size a[10],i,n variable.

Step 3: Read the size of array, Read n Value.

Step 4: For i=0 to i<n goto Step 5.

Step 5 :Read a[i].

Step 6: Calculate the sum value and average value.

Step 7: For i=0 to i<n goto Step 8.

Step 8: sum=sum+a[i],

average=sum/n.

Step 9: Print the Sum value.

Step 10: Print the average value.

Step 11:Stop.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int a[10],i,n;
  float sum=0, average;
  clrscr();
  printf("Enter the size of array:");
  scanf("%d", &n);
  printf("enter the elements:\n");
  for(i=0;i<n;i++)
  scanf("%d",&a[i]);
  for(i=0;i<n;i++)
  {
    sum=sum+a[i];
    average=sum/n;
  }
  printf(" Sum = %f\n",sum);
  printf("Average = %f\n",average);
  getch();
}
```

Output:

```
 Enter the size of array:5
enter the elements:
10
20
30
40
50
 Sum = 150.000000
Average = 30.000000
```

```
Enter the size of array:5
enter the elements:
10
20
30
40
50
 Sum = 150.000000
Average = 30.000000
```

Program 9: To remove duplicate element in single dimensional array.

Step 1: Start.

Step 2: Initialize the array size int a[50],I,j,k,dup[50],n;

Step 3: Read array n.

Step4: Repeat from i=0 to n.

Step5: Read a[i].

Step 6: Print the number of elements

Repeat until i=0 to n.

Step7: Repeat from j=i+1 to n.

Step 8: Repeat k=j to n.

a[k]=a[k+1]

j--, n--

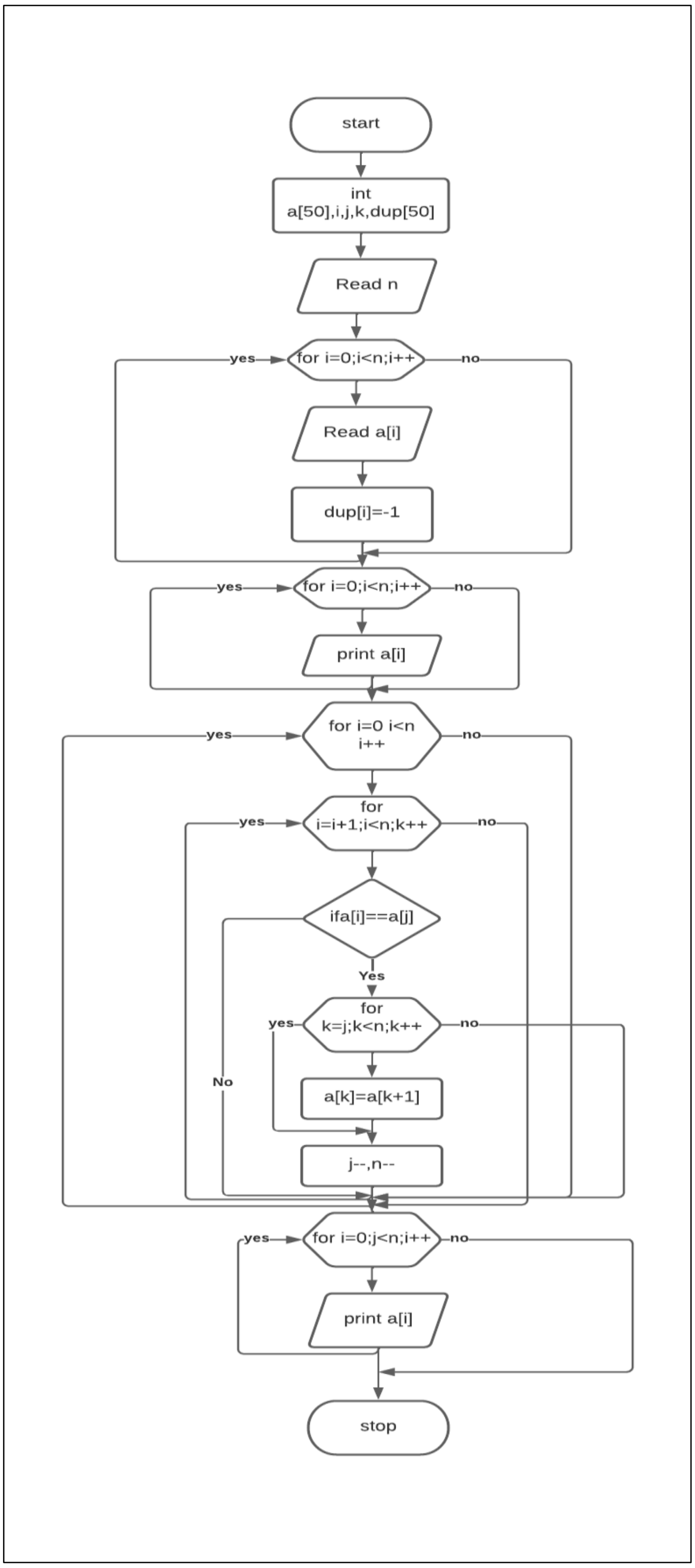
Step 9: End loops

Step 10: Repeat from i=0 to n.

Step 11: Print the elements after

deleting duplicate elements a[i].

Step 12: Stop.



```
#include<stdio.h>

#include<stdlib.h>

int main()

{


   int a[50],i,j,k, n;

   clrscr();

   printf("Enter size of the array\n");

   scanf("%d",&n);

   printf("Enter Elements of the array:\n");

   for(i=0;i<n;i++){

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

   }
   printf("Entered element are: \n");
   for(i=0;i<n;i++){
     printf("%d ",a[i]);
   }
   for(i=0;i<n;i++)
   {
     for(j =i+1;j<n; j++)
     {
          if(a[i] == a[j])
          {
            for(k=j; k<n; k++)
            {
              a[k] = a[k+1];
            }
            j--;
            n--;
          }
     }
   }
   printf("\nAfter deleting the duplicate element the Array is:\n");
   for(i=0;i<n;i++){
     printf("%d ",a[i]);
   }  getch();
 }
```

Output:

```
Enter size of the array
10
Enter Elements of the array:
10 20 30 20 40 50 60 70 50 20
Entered element are:
```

```
10 20 30 20 40 50 60 70 50 20
After deleting the
duplicate element the Array is:
10 20 30 40 50 60 70
```

Program 10: To perform addition and subtraction of two matrices.

```
#include<stdio.h>
#include<Conio.h>
void main()
{
int a[10][10],b[10][10],sum[10][10],diff[10][10],i,j,m, n;
clrscr();

printf("\n Enter the order of matrix :");
scanf("%d %d",&m, &n);

printf("\n Enter the elements of matrices a:\n");
for(i=0;i<m;i++)
for(j=0;j<n;j++)
scanf("%d",&a[i][j]);

printf("\n Entr the elements of matrices b:\n");
for(i=0;i<m;i++)
for(j=0;j<n;j++)
scanf("%d",&b[i][j]);

for(i=0;i<m;i++)
for(j=0;j<n;j++)
sum[i][j]=a[i][j]+b[i][j];
printf("\n The summation matrix : \n");
for(i=0;i<m;i++)
{
```

```
for(j=0;j<n;j++)
{
printf("%4d ",sum[i][j]);
}
printf("\n");
}
printf("\nThe differnce of Matrix:\n");
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
diff[i][j]=a[i][j]-b[i][j];
}
}
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
printf(" %4d",diff[i][j]);
}
printf("\n");
}
getch();
}
```

Output:

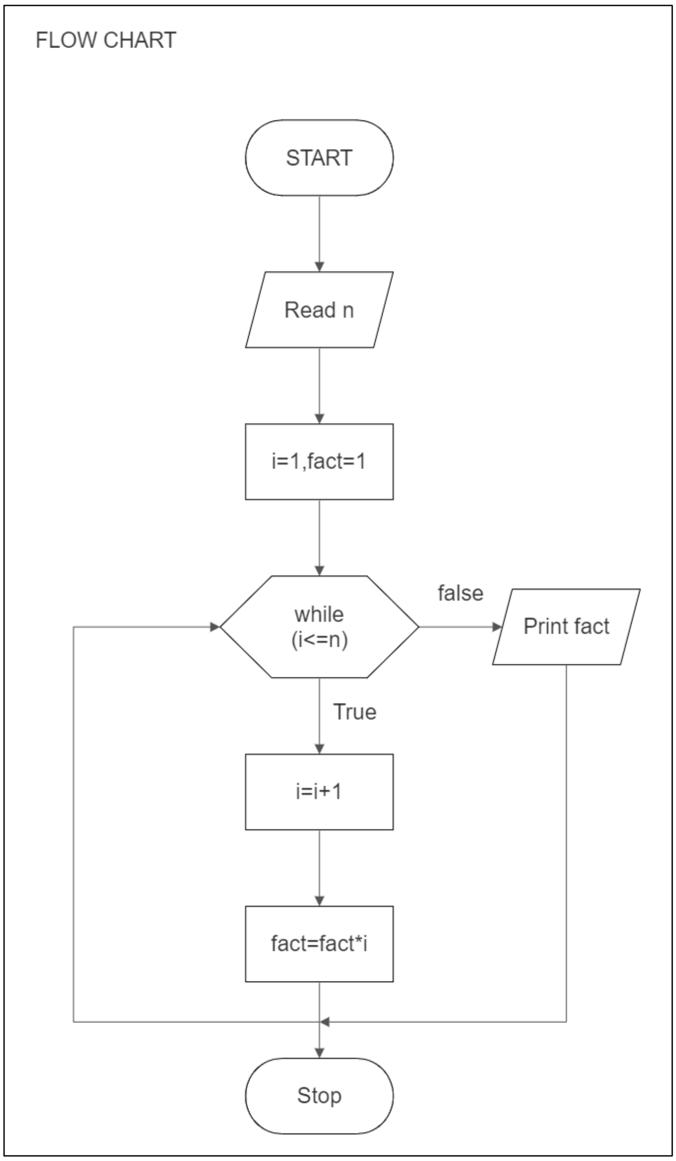
```
 Enter the order of matrix :2 2

 Enter the elements of matrices a:
1 2 3 4

 Entr the elements of matrices b:
1 2 3 4
```

```
 The summation matrix :
   2    4
   6    8
The differnce of Matrix:
    0    0
    0    0
```

Program 11: To find the Factorial of a number.

Algorithm:

```
Step 1: Start.
Step 2: Read the number N.
Step 3: Initialize the Variable i=1,fact=1.
Step 4: Repeat step 4 through 6 until i<=n
Step 5: fact=fact*i.
Step 6: i=i+1.
Step 7: Print the fact value.
Step 8: End.
```



```
#include<stdio.h>
#include<conio.h>
void main()
{
int fact=1,i=1,n;
clrscr();
printf("enter the  number :\n");
scanf("%d", &n);
while(i<=n)
{
fact=fact*i;
i++;
}
printf("Factorial num for %d is %d",n,fact);
getch();
  }
```

Output:

```
enter the  number : 5.
Factorial num for 5 is 120.
```

Program 12: To generate Fibonacci series

Algorithm:

Step 1: Start.

Step 2: Initialize the variable i=3,fib1=0,fib2=1,fib,n.

Step 3: Read the n value.

Step 4: for i=3 to i<=n   go to step 7.

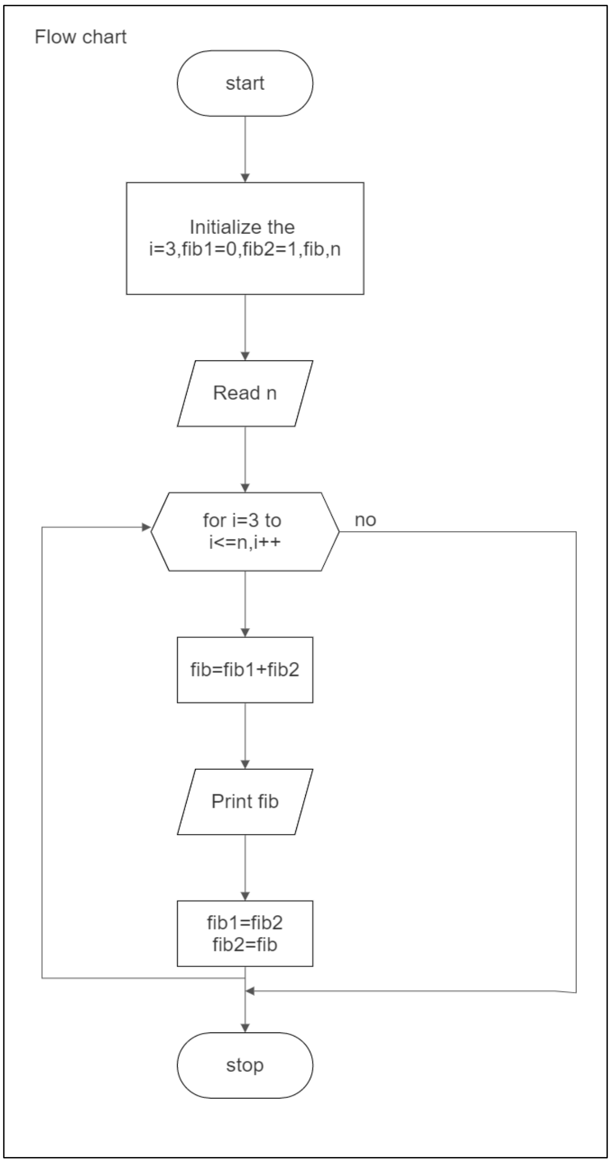
Step 5: fib1=fib1+fib2.

Step 6: Print the fib value.

Step 7: End for Statement.

Step 8: Stop..

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int i=3,fib1=0,fib2=1, fib, n;
  clrscr();
  printf("Enter the terms:");
  scanf("%d",&n);
  printf("Fibonacci series : %d  %d ",fib1,fib2);
  for(i=3;i<=n;i++)
  {
     fib=fib1+fib2;
     printf("  %d ",fib);

     fib1=fib2;
     fib2=fib;
  }
  getch();
```



```
}
```

Output:

```
Enter the terms:5
Fibonacci series : 0  1   1   2   3
Enter the terms:9
Fibonacci series : 0  1   1   2   3   5   8   13   21
```

Program 13: To find the length of a string without using a built in function.

Algorithm:

Step 1: Start.

Step 2: Initialize the Variable char name[20], int i=0,length.

Step 3: Read the name.

Step 4: If while name[i]!=’\0’ go to step 5, otherwise go to step 6.

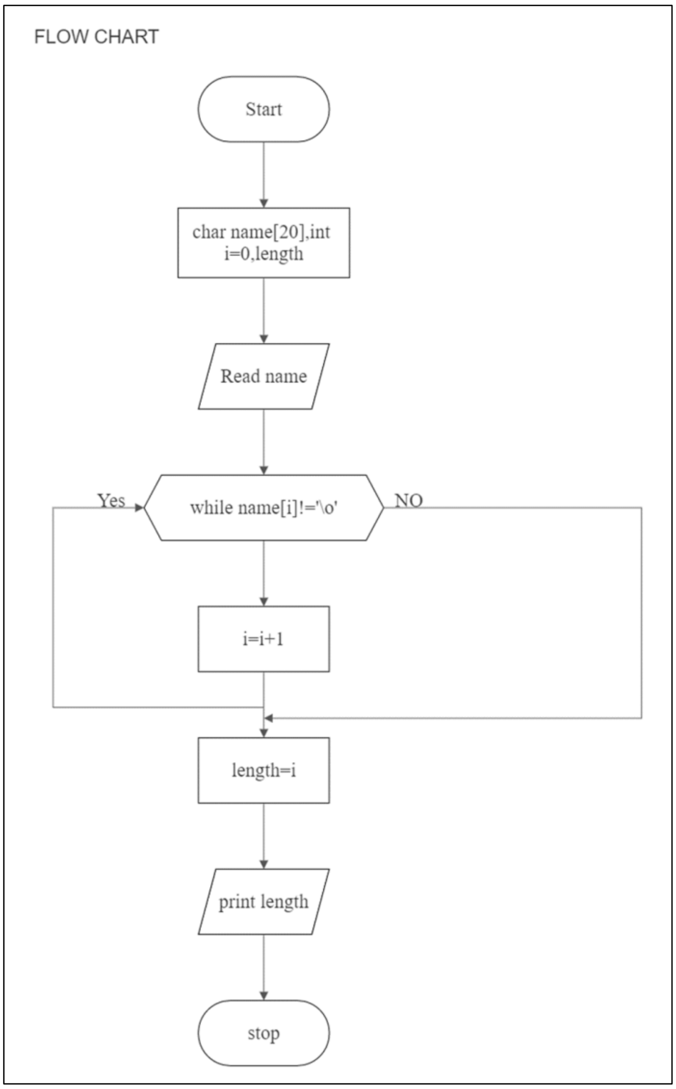
Step 5: Increment i value by 1,i=i+1.

Step 6: Assign the Value of i to length variable.length=1.

Step 7: Print the length of String.

Step 8: Stop.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char name[20];
int i=0, length;
clrscr();
printf("Enter the name:");
gets(name);
while(name[i]!=0)
{
i++;
}
```



```
length=i;
printf("length of the string is %d",length);
getch();
}
```

Output:

```
Enter the name:  KLEBCA
length of the string is 6
```

Program 14: program to demonstrate string functions.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
   char s1[20],s2[20];
   clrscr();
   printf("Enter string 1:");
   gets(s1);
   printf("Enter string 2:");
   gets(s2);
   printf("\n String1: %s \n string2 :%s",s1,s2);
   printf("\nLength of %s: %d",s1,strlen(s1));
   printf("\nString copy in string2: %s",strcpy(s2,"Nagarbhavi"));;
   printf("\nString concatination: %s",strcat(s1,s2));
   printf("\nCompare string1 and String2: %d",strcmp(s1,"kle"));
   printf("\nString in lowercase: %s",strlwr(s1));
   printf("\nstring in uppercase: %s",strupr(s2));
   printf("\nString Reverse: %s",strrev(s1));
   getch();
}
```

```
Enter string 1:kle
Enter string 2:bca


 String1: kle

 string2 :bca

Length of kle: 3

String copy in string2: Nagarbhavi

String concatination: kleNagarbhavi

Compare string1 and String2: 78

String in lowercase: klenagarbhavi

string in uppercase: NAGARBHAVI

String Reverse: ivahbraganelk
```

Program 15: To read,display and add m*n matrices using function.

```
#include<stdio.h>

#include<conio.h>

void add(int m,int n,int a[10][10],int b[10][10])

{

   int i,j,sum[10][10];

   for(i=0;i<m;i++)

     for(j=0;j<n;j++)

            sum[i][j]=a[i][j]+b[i][j];

            printf("\n Sum of the given matrices:\n");

            for(i=0;i<m;i++)

            {

               for(j=0;j<n;j++)

               printf(" %d ",sum[i][j]);

               printf("\n");

            }

}

void display(int m,int n,int a[10][10],int b[10][10])
```

```
{
   int i,j;
   printf("Matrix A:\n");
    for(i=0;i<m;i++)
           {
             for(j=0;j<n;j++)
             printf(" %d ",a[i][j]);
             printf("\n");
           }
    printf("Matrix B:\n");
    for(i=0;i<m;i++)
           {
             for(j=0;j<n;j++)
             printf(" %d ",b[i][j]);
             printf("\n");
           }
}
void read(int m,int n,int a[10][10],int b[10][10])
{
  int i,j;
  printf("Enter %d elements of the first matrix row:\n",m*n);
  for(i=0;i<m;i++)
    for(j=0;j<n;j++)
          scanf("%d",&a[i][j]);
  printf("Enter %d elements of the second matrix row:\n",m*n);
  for(i=0;i<m;i++)
    for(j=0;j<n;j++)
          scanf("%d",&b[i][j]);
}
void main()
{
    int m,n,i,j,a[10][10],b[10][10],sum[10][10];
    clrscr();
    printf("Enter the order of matrices::");
    scanf("%d %d",&m,&n);
    read(m,n,a,b);
    display(m,n,a,b);
    add(m,n,a,b);
    getch();
}
```

Output:

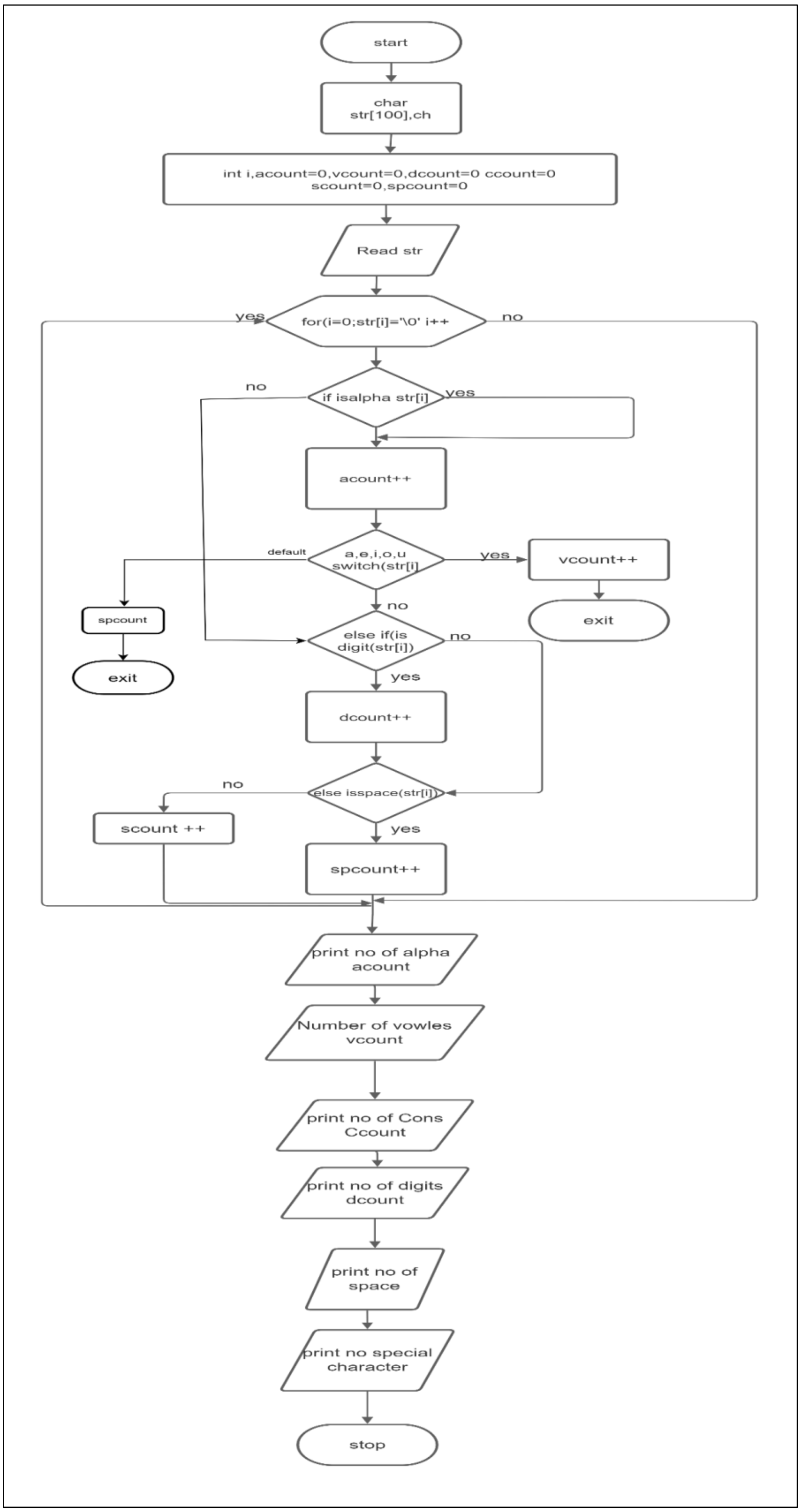
```
Enter the order of matrices::2 3
Enter 6 elements of the first matrix row:
1 2 3
4 5 6
Enter 6 elements of the second matrix row:
1 2 3
4 5 6
Matrix A:
 1  2  3
 4  5  6
Matrix B:
 1  2  3
 4  5  6

 Sum of the given matrices:
 2  4  6
 8  10  12
```

Program 16: To read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters:

Algorithm:

```
Step 1:Start.
Step 2: Initialization Variable, Char str[100],ch,int I,int acount=0,vcount=0,
Dcount=0,ccount=0,scount=0,spcount=0.
Step 3: Read the String.
Step 4: Repeat for i=0 to str[i]!='\0'
        Check  If string is alphabet increment account.
        Else If string contains vowels increment vowels else increment ccount.
        Else if string is numeric increment dcount.
        Else if string contains special character and spaces increment scount and spcount.
Step 5: print the number of alphabets, vowels, consonents, digits, spaces and special
character.
Step 6: End.
```

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
void main()
{
   char str[100],ch;
   int i,acount=0, vcount=0,dcount=0,ccount=0,scount=0,spcount=0;
   clrscr();


  printf("\n Enter the string:");
   gets(str);
   for(i=0;str[i]!='\0';i++)
   {   if(isalpha(str[i]))
  {
  acount++;
  switch(str[i])
       {
         case 'a':
         case 'e':
         case 'i':
        case 'o':
        case 'u':
        vcount++;
        break;


        default:ccount++;
       }
       }
else if(isdigit(str[i]))
       dcount++;
 else if(isspace(str[i]))
       scount++;
else
       spcount++;


}
printf("\n Number of Alphabets= %d",acount);
```



```
    printf("\n Number of vowels=%d",vcount);

    printf("\n Number of Consonents=%d",ccount);

    printf("\n Number of Digits=%d",dcount);

    printf("\n Number of Spaces=%d",scount);

    printf("\n Number of special symbols=%d",spcount);

    getch();

}
```

```
 Enter the string:KLE BCA Nagarbhavi @560096

 Number of Alphabets= 16
 Number of vowels=4
 Number of Consonents=12
 Number of Digits=6
 Number of Spaces=3
 Number of special symbols=1
```

Program 17: To swap two numbers using pointers

Algorithm:

Step 1: Initialize the Variable x,y.

Step 2: Read the x and y.

Step 3: Print the x,y Value.

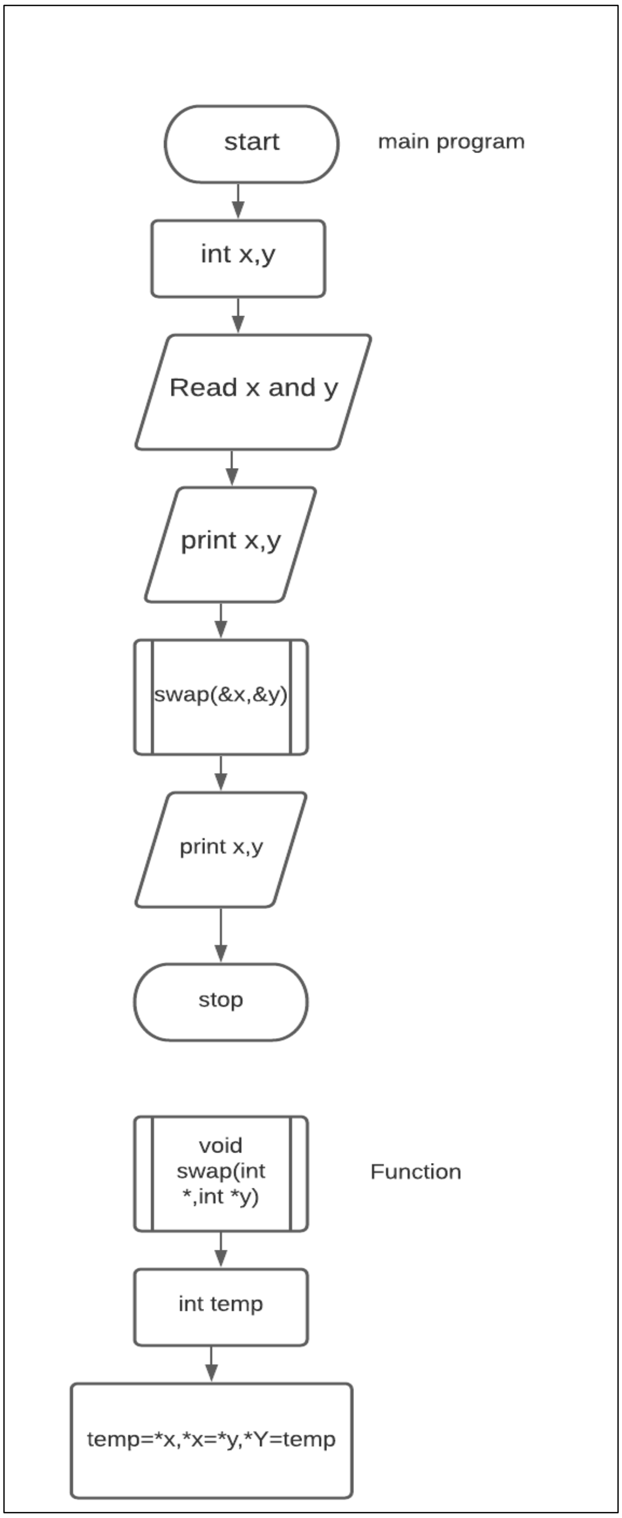
Step 4: Call the swap(&x,&y) function.

Step 5: In the swap function (int *x,int *y)swap the values. Initialize
The temp variable.

Step 6: Exchange the Variable temp=*x,*x=*y,*y=temp.

Step 7: Stop the Program.

```
#include<stdio.h>
#include<conio.h>
void swap(int *p1,int *p2);
void main()
{
   int x,y;
   clrscr();
   printf("enter the value of x and y");
   scanf("%d %d",&x,&y);
   printf("\n Before calling swap: x=%d y=%d",x,y);
   swap(&x,&y);
   printf("\n After calling swap: x=%d y=%d",x,y);
   getch();

}
void swap(int *x,int *y)
{
   int temp;
   temp=*x;
   *x=*y;
   *y=temp;
}
```



Output:

```
Enter the value of x and y 20 10

 Before calling swap: x=20 y=10
After calling swap: x=10 y=20
```

Program 18: To demonstrate student structure to read and display records of n students.

Algorithm:

Step 1: Start.

Step 2: Create the Structure name as Student.

Step 3: Initialize the variable regno,name[20],grade[2]n,i and struct student std[50].

Step 4: Read the number of students. Read n value.

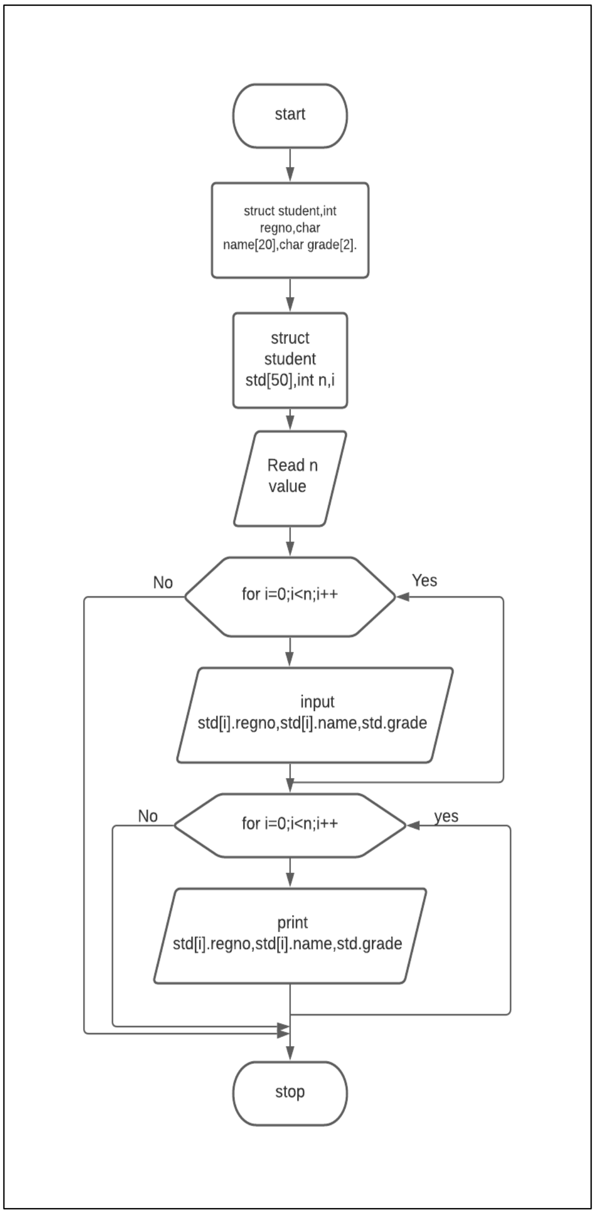
Step 5: Repeat for i=0 to n.

Step 6: Read the regno, name and grade of an student

Step 7: Repeat for i=0 to n,

Step 8: Print the student Regno, name and grade.

Step 9: Stop the Program.


start
struct student,int regno,char name[20],char grade[2].
struct student std[50],int n,i
Read n value
No
for i=0;i<n;i++
Yes
input std[i].regno,std[i].name,std.grade
No
for i=0;i<n;i++
yes
print std[i].regno,std[i].name,std.grade
stop

```
#include<stdio.h>
#include<conio.h>
struct student
{
   int regno;
   char name[20];
   char grade[2];
};
void main()
{
   struct student std[50];
   int n,i;
   clrscr();
   printf("Enter the number of students:");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
       printf("Enter the regno name and grade of Student %d:\n",i+1);
       scanf("%d %s %s",&std[i].regno,std[i].name,std[i].grade);
```

```
  }
  printf("\n Regno \t Name \t Grade");
  for(i=0;i<n;i++)
  {
     printf("\n %d \t %s \t %s",std[i].regno,std[i].name,std[i].grade);
  }
  getch();


}
```

Output:

```
Enter the number of students:3
Enter the regno name and grade of Student 1:
11 ABC A
Enter the regno name and grade of Student 2:
22 XYZ B
Enter the regno name and grade of Student 3:
33 EFG C

 Regno   Name    Grade
 11      ABC     A
 22      XYZ     B
 33      EFG     C
```

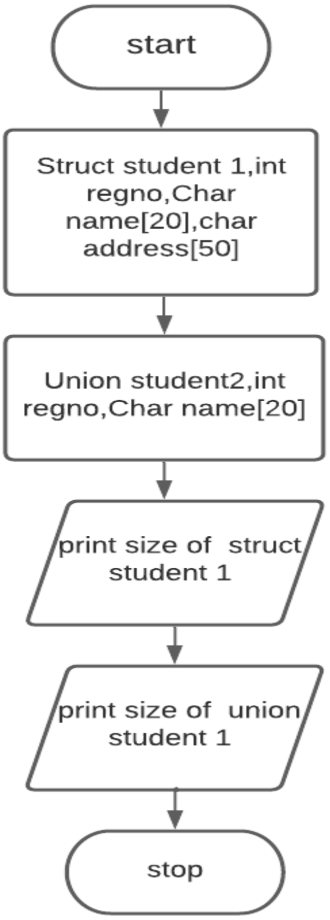
Program 19: To demonstrate the difference between structure and union.

Algorithm:

```
Step 1: Start.
Step 2: Create the Structure as Student1.
Step3: Initialize the variable regno, name, address.
Step4: Create the union, Student2.
Step5: Initialize the variable is regno, name, address.
```

Step6: Using size of keyword print the size of Structure and size of union .

Step 7: Stop.


start
Struct student 1,int regno,Char name[20],char address[50]
Union student2,int regno,Char name[20]
print size of struct student 1
print size of union student 1
stop

```
#include<stdio.h>
#include<conio.h>
struct student1
{
   int regno;
   char name[20];
   char address[50];
};

union student2
{
   int regno;
   char name[20];
   char address[50];
};
void main()
{
   clrscr();
   printf("\n Size of structure=%d",sizeof(struct student1));
   printf("\n Size of union=%d",sizeof(union student2));
   getch();
}
```

Output:

```
 Size of structure=72
 Size of union=50
```