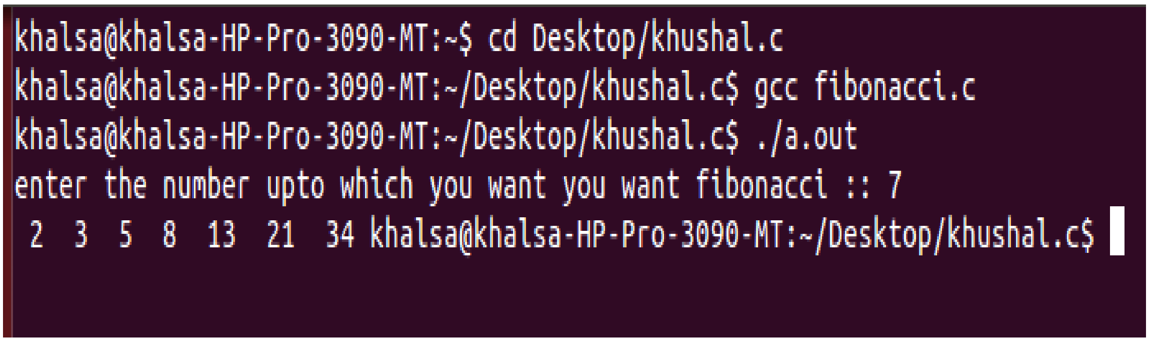
C PROGRAMMING & DATA STRUCTURES

KHUSHAL KAPOOR
2015ELE1020
BSc(H) Electronics

Fibonacci Series

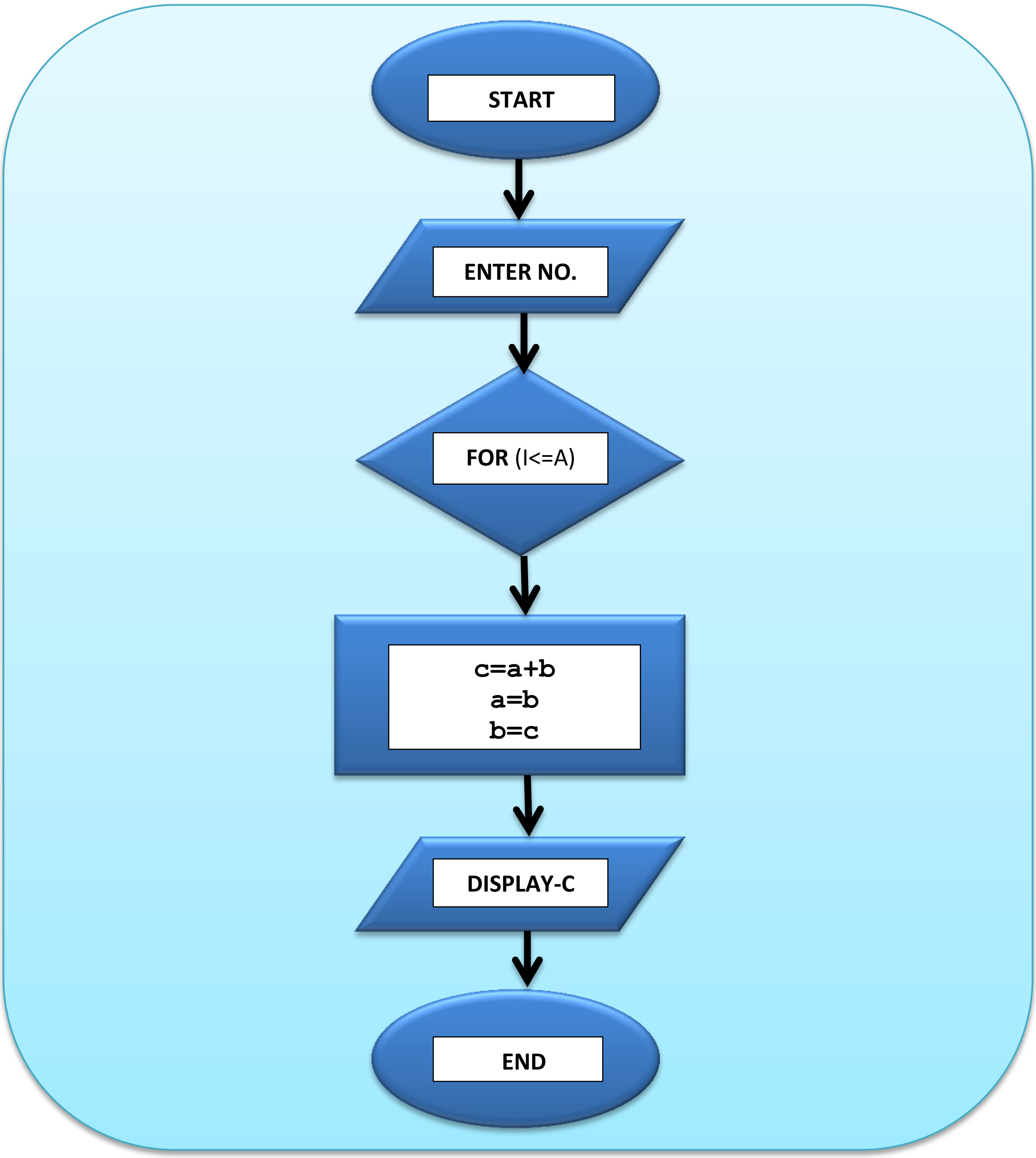
```
#include<stdio.h>
void main()
{
      int a=1,b=1,n,c,i;
      printf("enter the number upto which you want you want
fibonacci :: ");
      scanf("%d",&n);
      for(i=1;i<=n;i++)
      {
            c=a+b;
            a=b;
            b=c;
            printf(" %d ",c);
      }
}
```

OUTPUT ::



```
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc fibonacci.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
enter the number upto which you want you want fibonacci :: 7
 2  3  5  8  13  21  34 khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ 
```

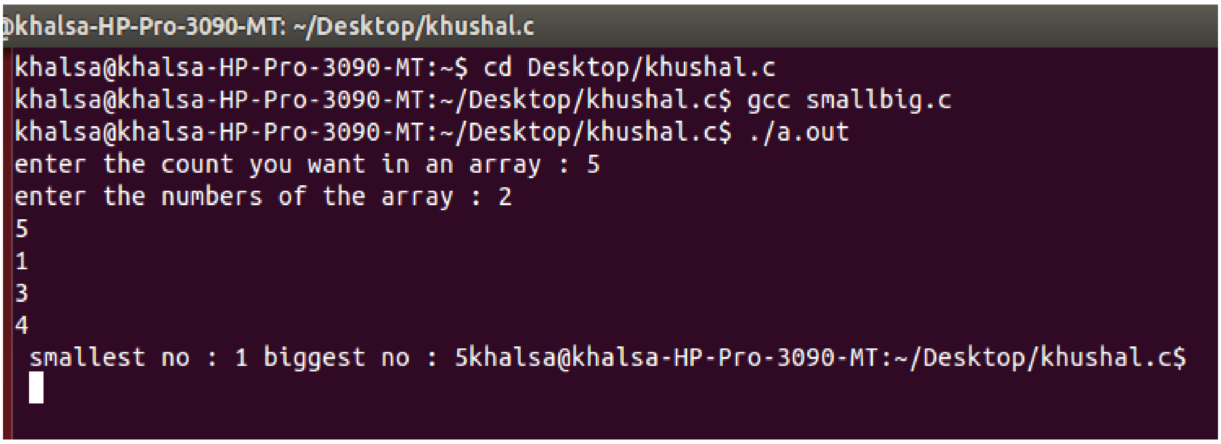
FLOW CHART ::



MINIMUM & MAXIMUM OF N NUMBERS

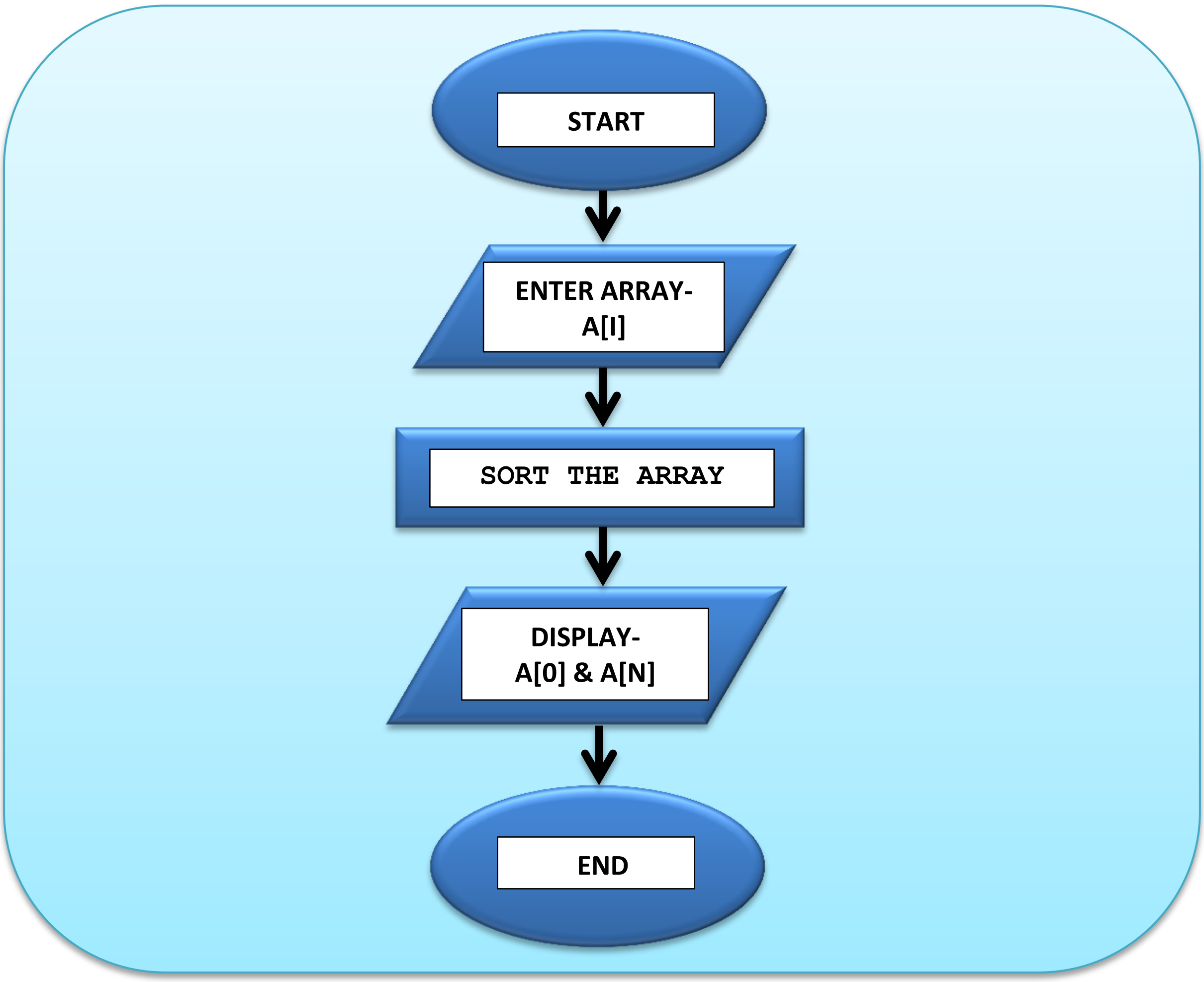
```
#include<stdio.h>
void main()
{
      int a[100],n,i,j,b;
      printf("enter the count you want in an array : ");
      scanf("%d",&n);
      printf("enter the numbers of the array : ");
      for(i=1;i<=n;i++)
      {
            scanf("%d",&a[i]);
      }
      for(i=1;i<=n;i++)
      {
            for(j=1;j<=n;j++)
            {
                  if(a[j]>a[j+1])
                  {
                        b=a[j];
                        a[j]=a[j+1];
                        a[j+1]=b;
                  }
            }
      }
      printf(" smallest no : %d",a[1]);
      printf(" biggest no : %d",a[n]);

}
```

OUTPUT ::



```
@khalsa-HP-Pro-3090-MT: ~/Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc smallbig.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
enter the count you want in an array : 5
enter the numbers of the array : 2
5
1
3
4
 smallest no : 1 biggest no : 5khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$
```

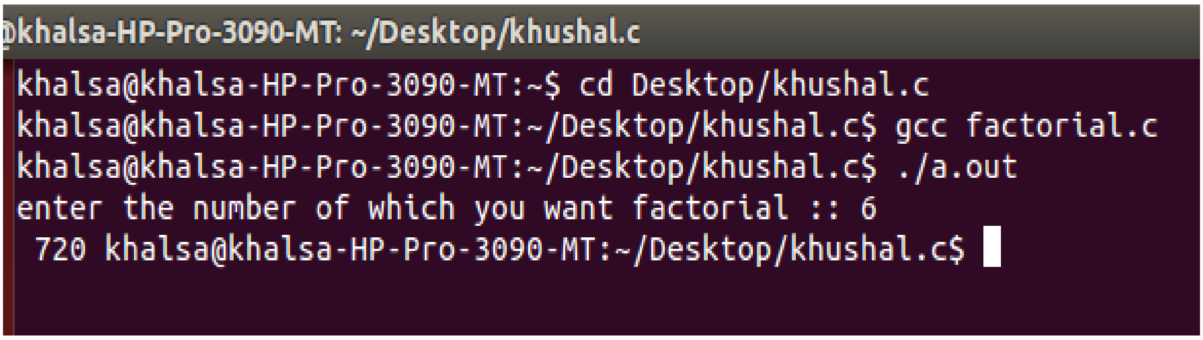
FLOW CHART ::



FACTORIAL

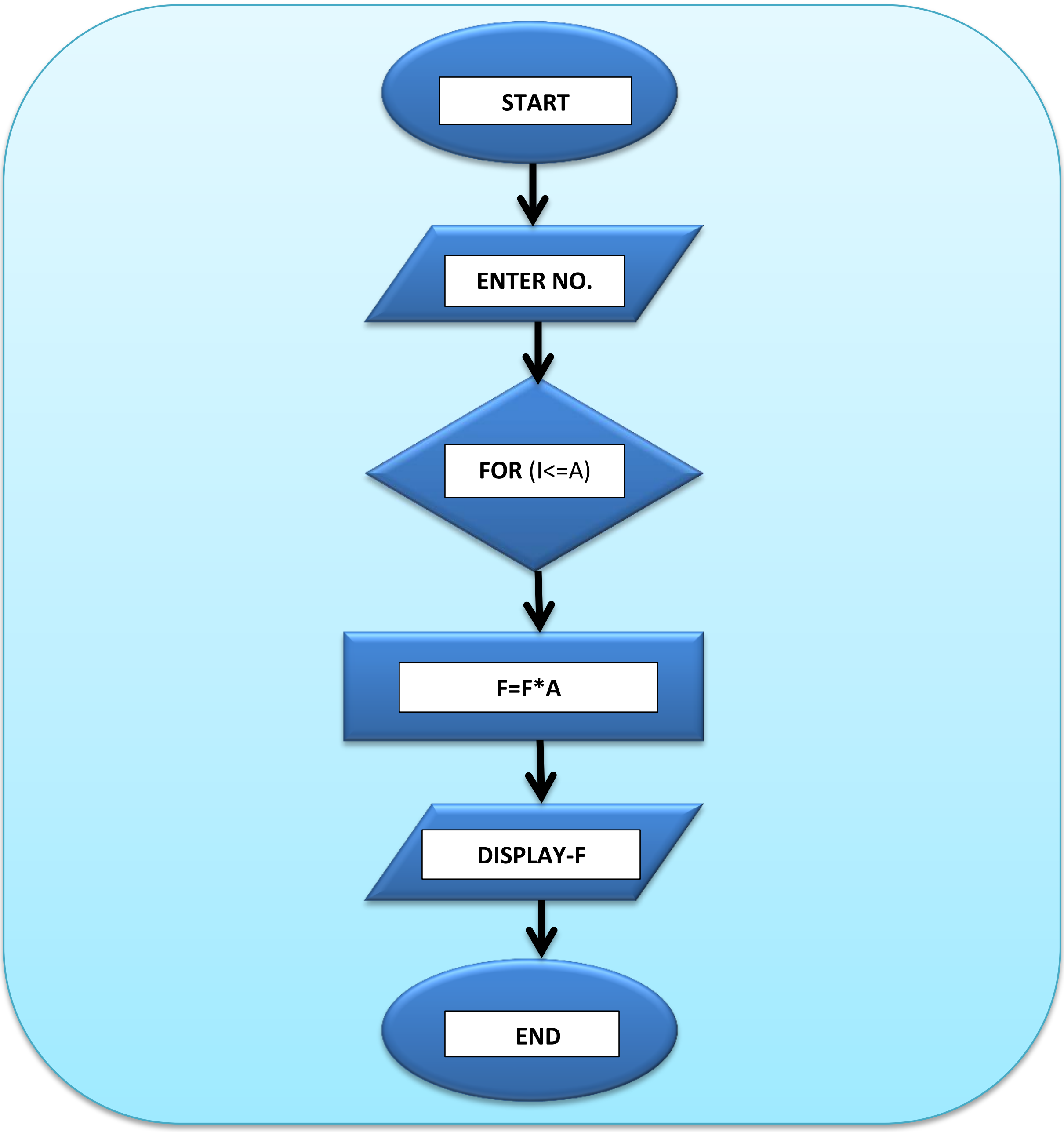
```
#include<stdio.h>
void main()
{
      int a,i,f=1;
      printf("enter the number of which you want factorial :: ");
      scanf("%d",&a);
      for(i=1;i<=a;i++)
      {
            f=f*i;
      }
      printf(" %d ",f);
}
```

OUTPUT ::



```
@khalsa-HP-Pro-3090-MT: ~/Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc factorial.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
enter the number of which you want factorial :: 6
 720 khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ 
```

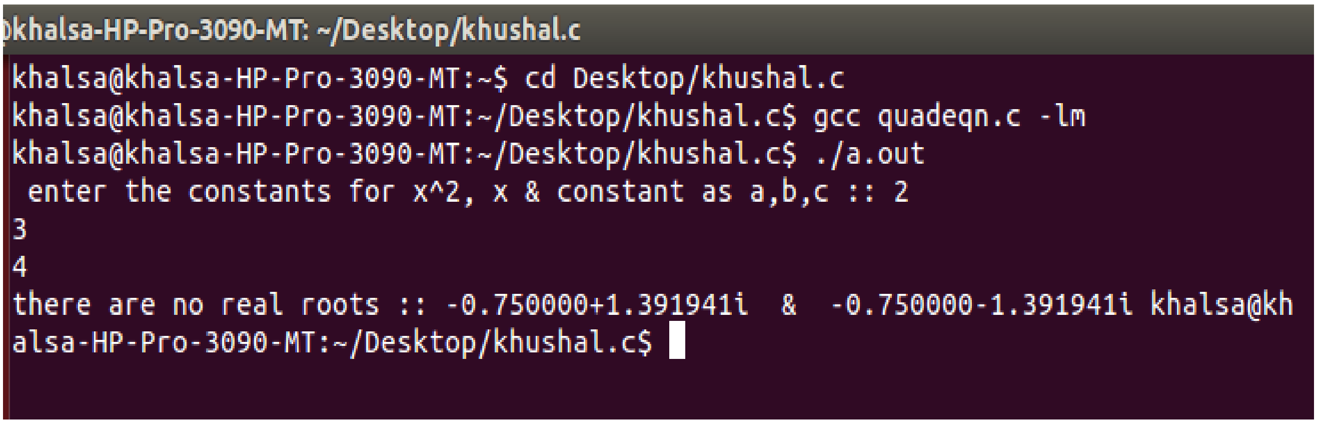
FLOW CHART ::



ROOTS OF QUADRATIC EQUATION

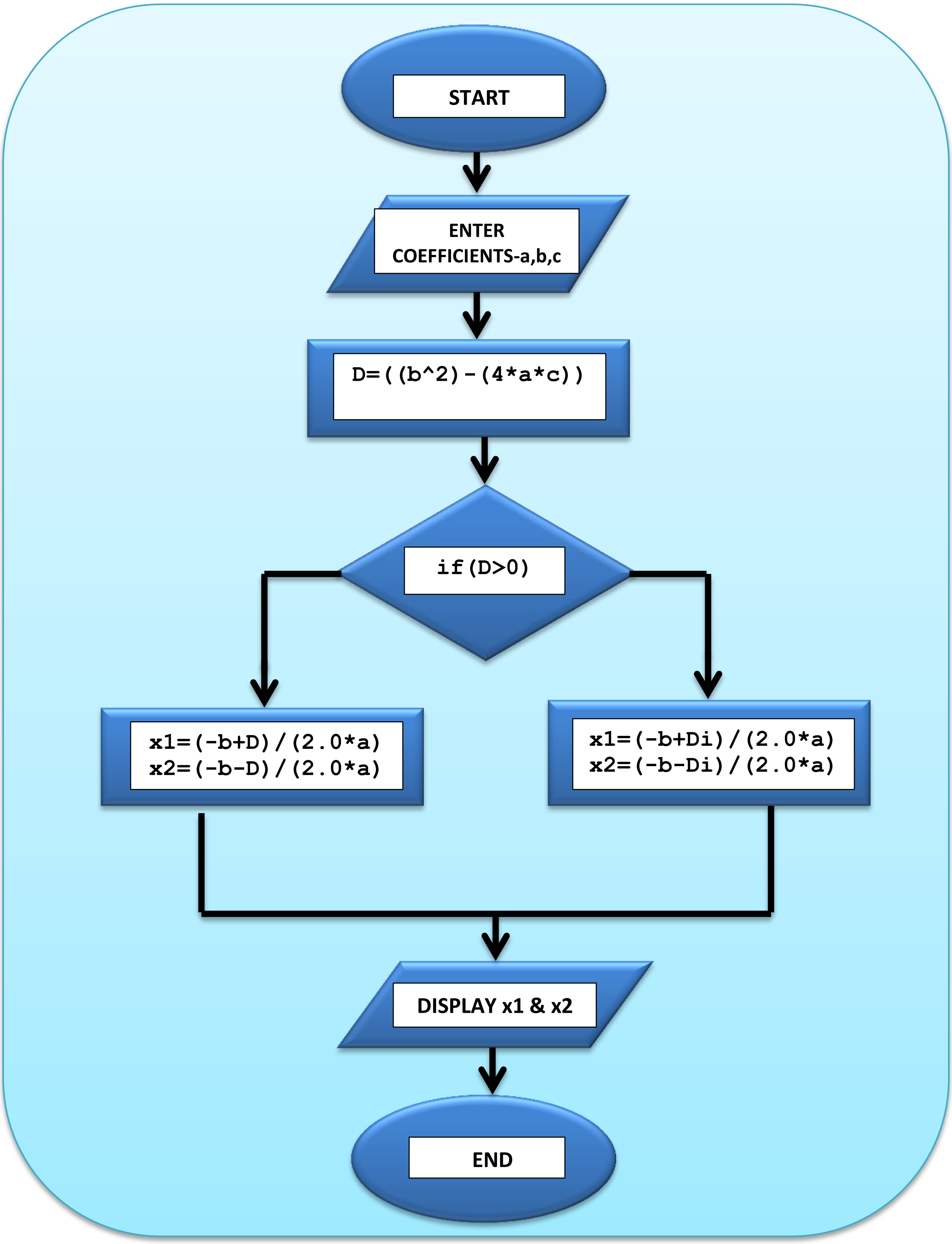
```
#include<stdio.h>
#include<math.h>
void main()
{
      int a,b,c;
      float D,x1,x2,x,d;
      printf(" enter the constants for x^2, x & constant as a,b,c :: 
");
      scanf("%d%d%d",&a,&b,&c);
      d=((b^2)-(4*a*c));
      if(d>0)
      {
            D=sqrt(d);
            x1=(-b+D)/(2.0*a);
            x2=(-b-D)/(2.0*a);
            printf("there were two real roots ::%f & %f",x1,x2);
      }
      else if(d<0)
      {
            D=sqrt(-d);
            printf("there are no real roots :: %f+%fi  &  %f%fi ",(-
b/(2.0*a)),(D/(2.0*a)),(-b/(2.0*a)),-(D/(2.0*a)));
      }
      else
      {
            x=-b/(2*a);
            printf("there is only one root :: %f",x);
      }
}
```

OUTPUT ::



```
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc quadeqn.c -lm
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
 enter the constants for x^2, x & constant as a,b,c :: 2
3
4
there are no real roots :: -0.750000+1.391941i  &  -0.750000-1.391941i khalsa@kh
alsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ 
```

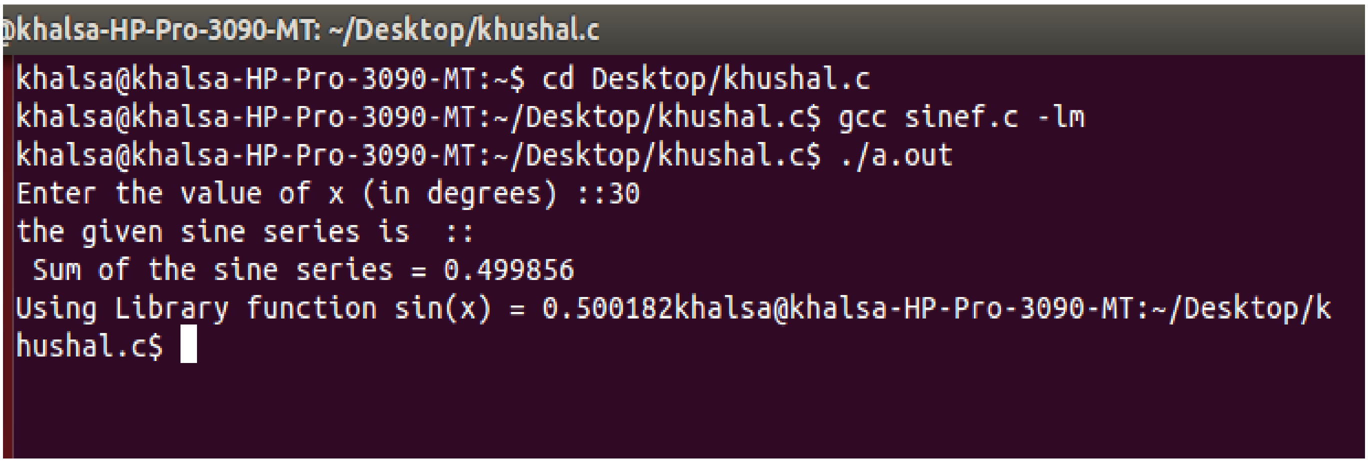
FLOW CHART ::


START
ENTER COEFFICIENTS-a,b,c
D=((b^2)-(4*a*c))
if(D>0)
x1=(-b+D)/(2.0*a)
x2=(-b-D)/(2.0*a)
x1=(-b+Di)/(2.0*a)
x2=(-b-Di)/(2.0*a)
DISPLAY x1 & x2
END

VALUE OF SIN(X)

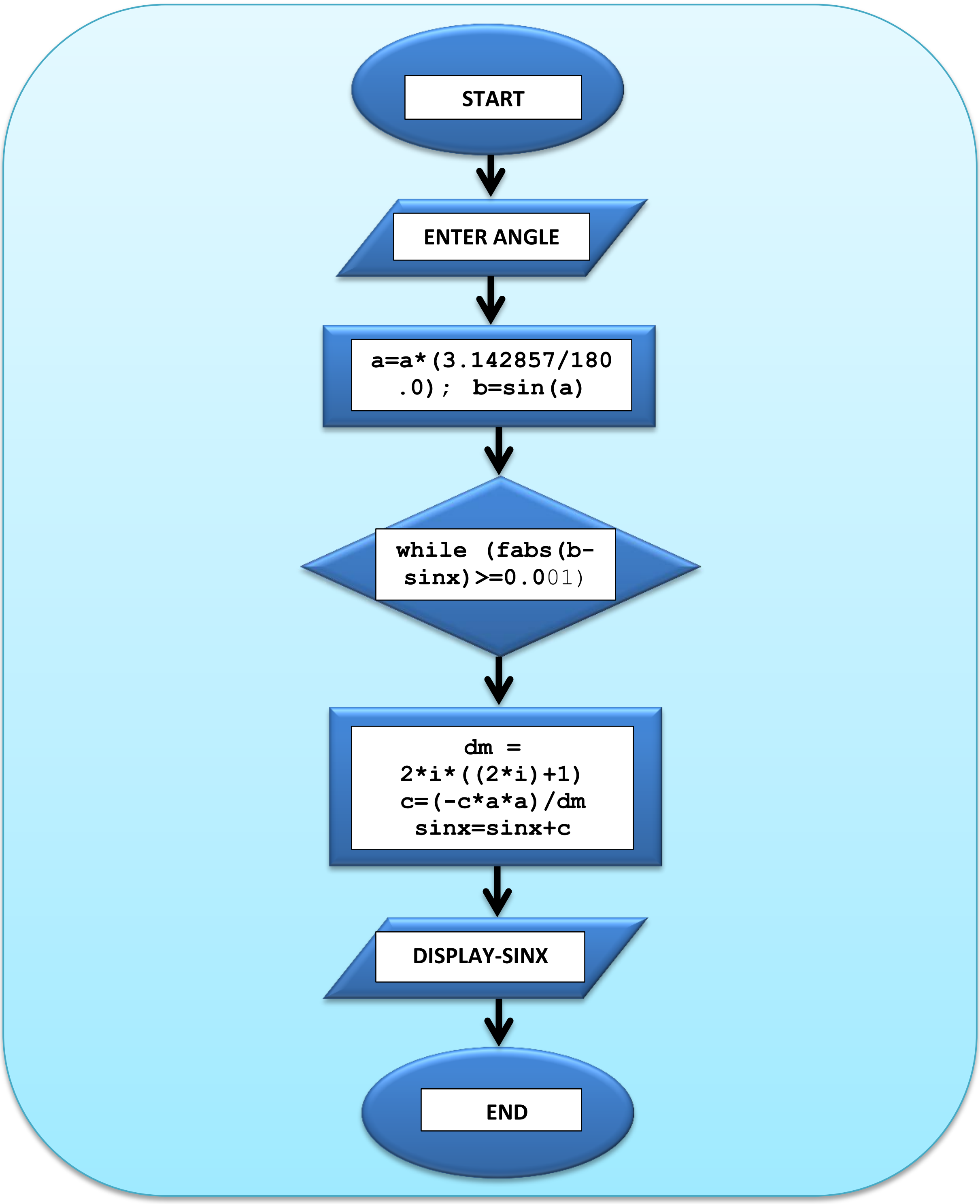
```
#include<stdio.h>
#include<math.h>
void main()
{
      float a,b,c,sinx,dm;
      int i=1;
      printf("Enter the value of x (in degrees) ::");
      scanf("%f",&a);
      a=a*(3.142857/180.0);
      b=sin(a);
      c=a;
      sinx=c;
      while (fabs(b-sinx)>=0.001)
      {
            dm = 2*i*((2*i)+1);
                  c=(-c*a*a)/dm;
            sinx=sinx+c;
            i=i++;
      }
      printf("the given sine series is  :: \n ");
      printf("Sum of the sine series = %f \n", sinx);
      printf("Using Library function sin(x) = %f", b);
}
```

OUTPUT ::



```
@khalsa-HP-Pro-3090-MT: ~/Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc sinef.c -lm
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
Enter the value of x (in degrees) ::30
the given sine series is  ::
 Sum of the sine series = 0.499856
Using Library function sin(x) = 0.500182khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/k
hushal.c$ 
```

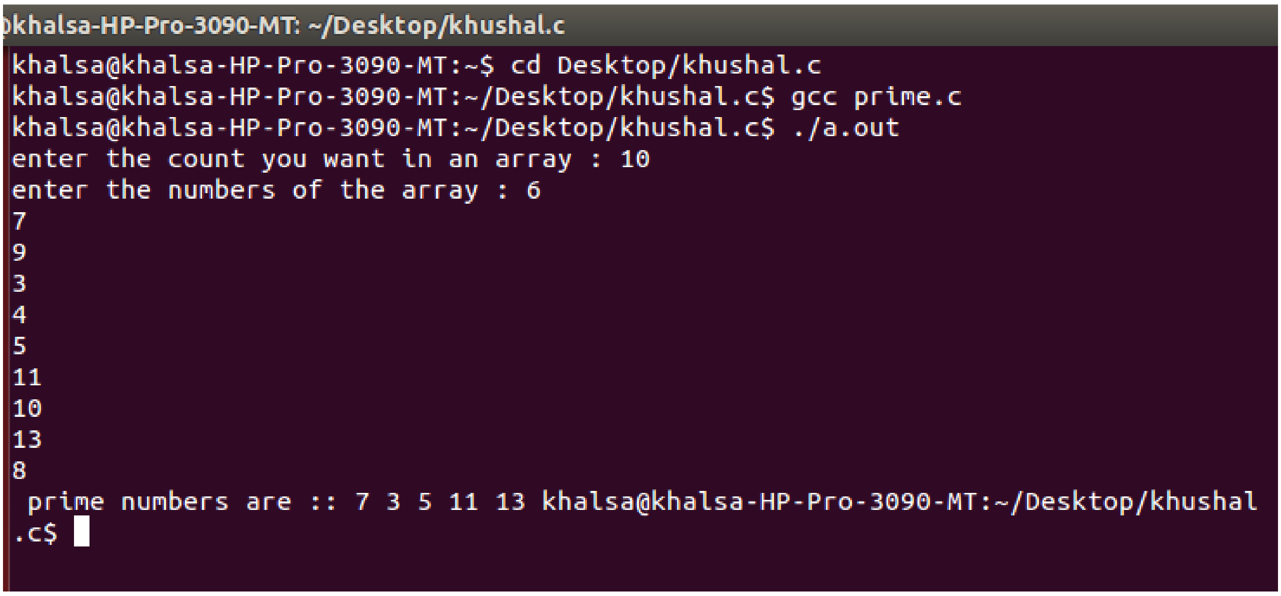
FLOW CHART ::


START
ENTER ANGLE
a=a*(3.142857/180.0); b=sin(a)
while (fabs(b-sinx)>=0.001)
dm =
2*i*((2*i)+1)
c=(-c*a*a)/dm
sinx=sinx+c
DISPLAY-SINX
END

PRIME NUMBERS

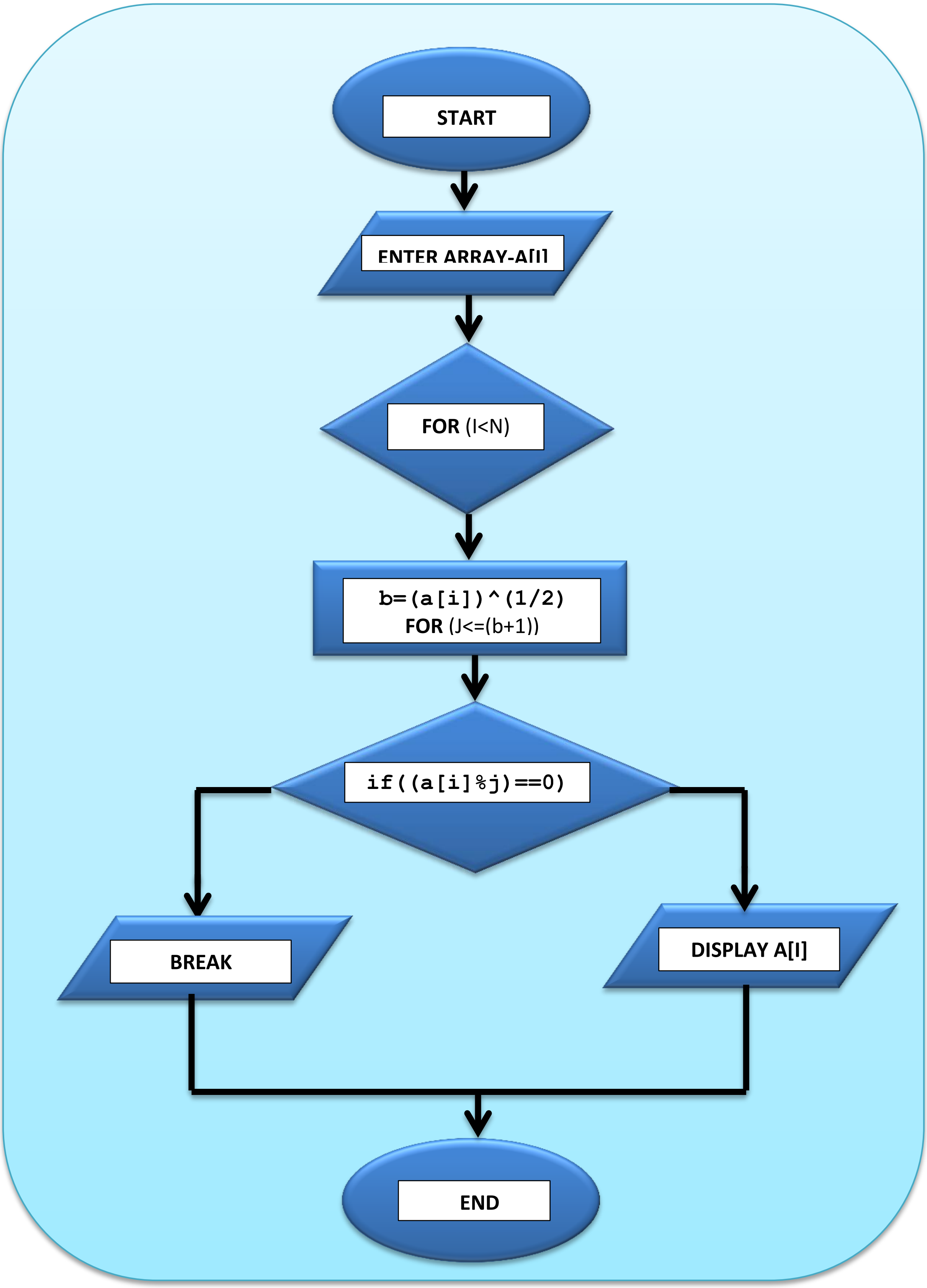
```
#include<stdio.h>
#include<math.h>
void main()
{
      int a[100],n,i,j;
      double b;
      printf("enter the count you want in an array : ");
      scanf("%d",&n);
      printf("enter the numbers of the array : ");
      for(i=1;i<=n;i++)
      {
            scanf("%d",&a[i]);
      }
      printf(" prime numbers are :: ");
      for(i=1;i<=n;i++)
      {
            int x=0;
            b=(a[i])^(1/2);
            for(j=2;j<=(b+1);j++)
                  if((a[i]%j)==0)
                        break;
            if(j>=b)
            printf("%d ",a[i]);
      }
}
```

OUTPUT ::



```
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc prime.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
enter the count you want in an array : 10
enter the numbers of the array : 6
7
9
3
4
5
11
10
13
8
 prime numbers are :: 7 3 5 11 13 khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal
.c$ 
```

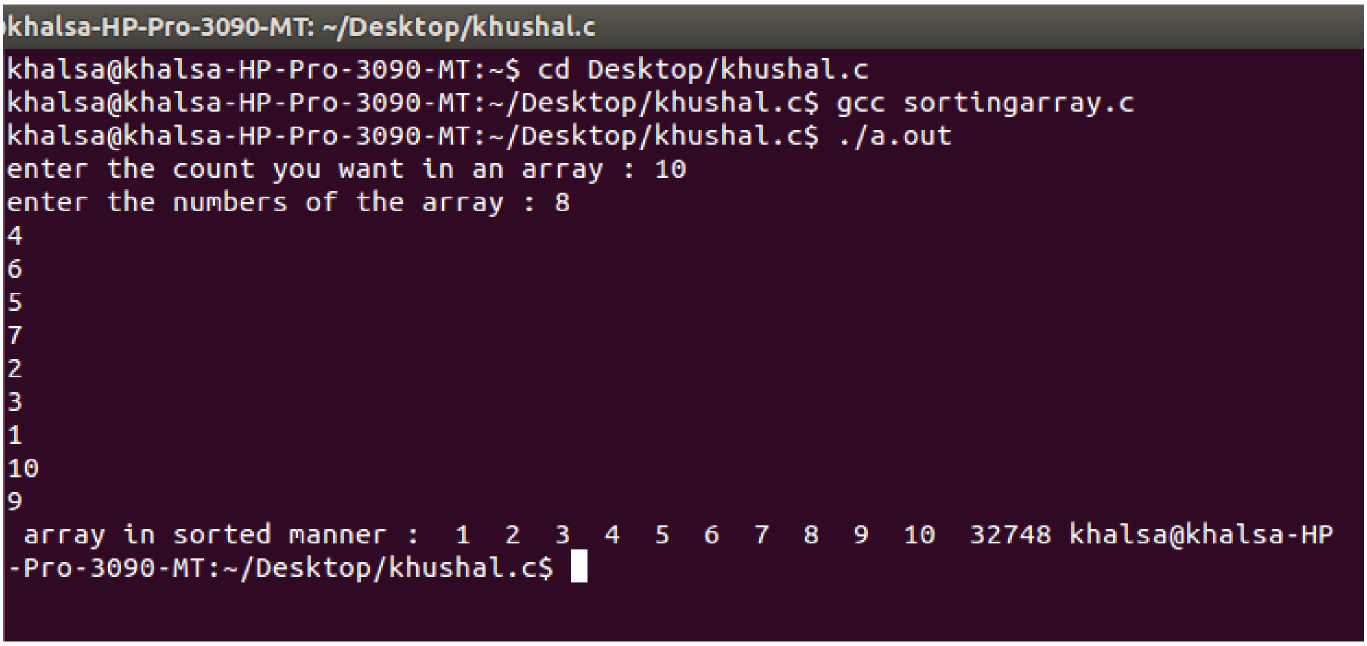
FLOW CHART ::


START
ENTER ARRAY-A[I]
FOR (I<N)
b=(a[i])^(1/2)
FOR (J<=(b+1))
if((a[i]%j)==0)
BREAK
DISPLAY A[I]
END

NUMBERS IN ASCENDING ORDER

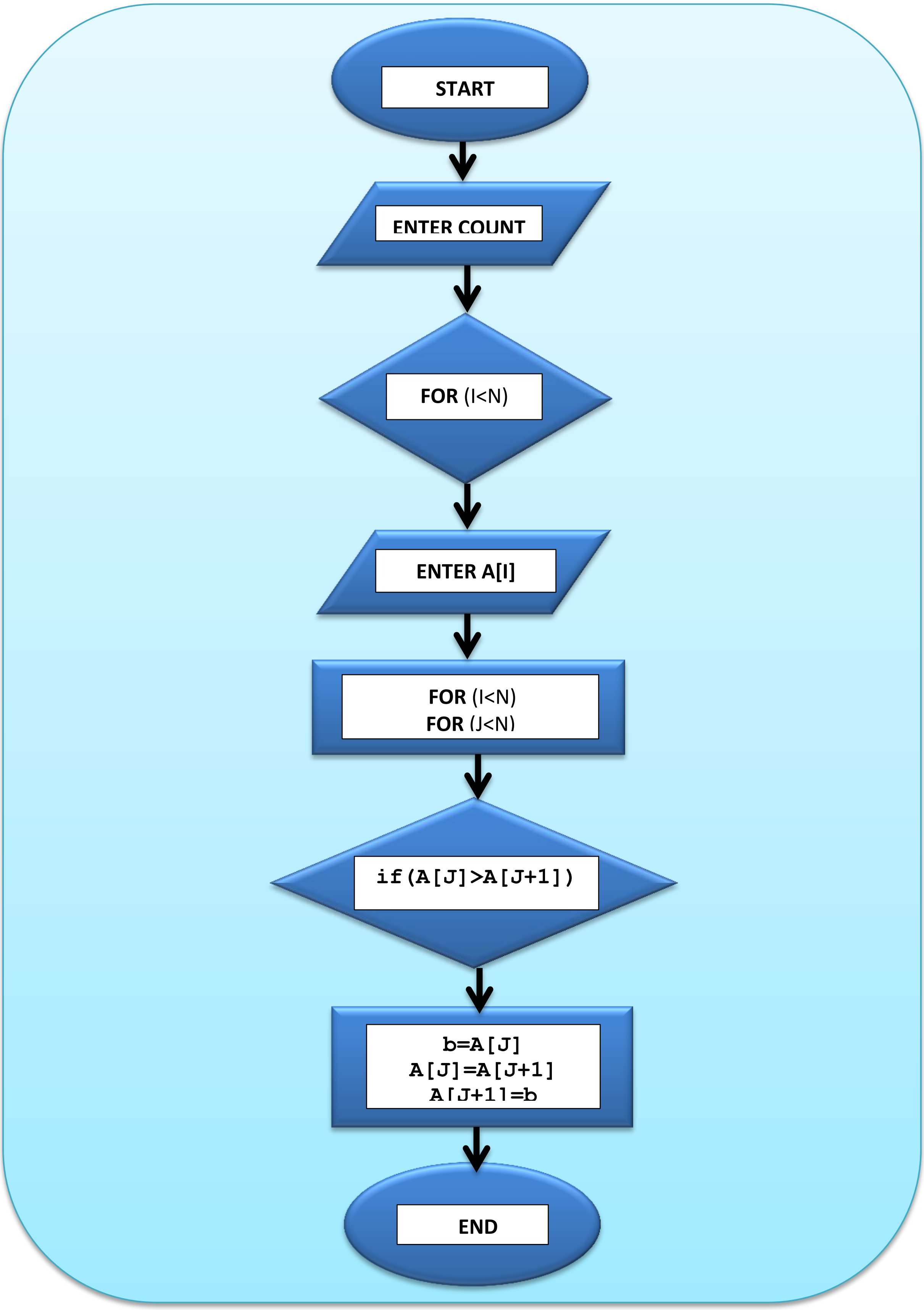
```
#include<stdio.h>
void main()
{
	int a[100],n,i,j,b;
	printf("enter the count you want in an array : ");
	scanf("%d",&n);
	printf("enter the numbers of the array : ");
	for(i=1;i<=n;i++)
	{
		scanf("%d",&a[i]);
	}
	for(i=1;i<=n;i++)
	{
		for(j=1;j<=n;j++)
		{
			if(a[j]>a[j+1])
			{
				b=a[j];
				a[j]=a[j+1];
				a[j+1]=b;
			}
		}
	}
	printf(" array in sorted manner : ");
	for(i=1;i<=(n+1);i++)
	{
		printf(" %d ",a[i]);
	}

}
```

OUTPUT ::



```
khalsa-HP-Pro-3090-MT: ~/Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc sortingarray.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
enter the count you want in an array : 10
enter the numbers of the array : 8
4
6
5
7
2
3
1
10
9
 array in sorted manner :  1  2  3  4  5  6  7  8  9  10  32748 khalsa@khalsa-HP
-Pro-3090-MT:~/Desktop/khushal.c$ 
```

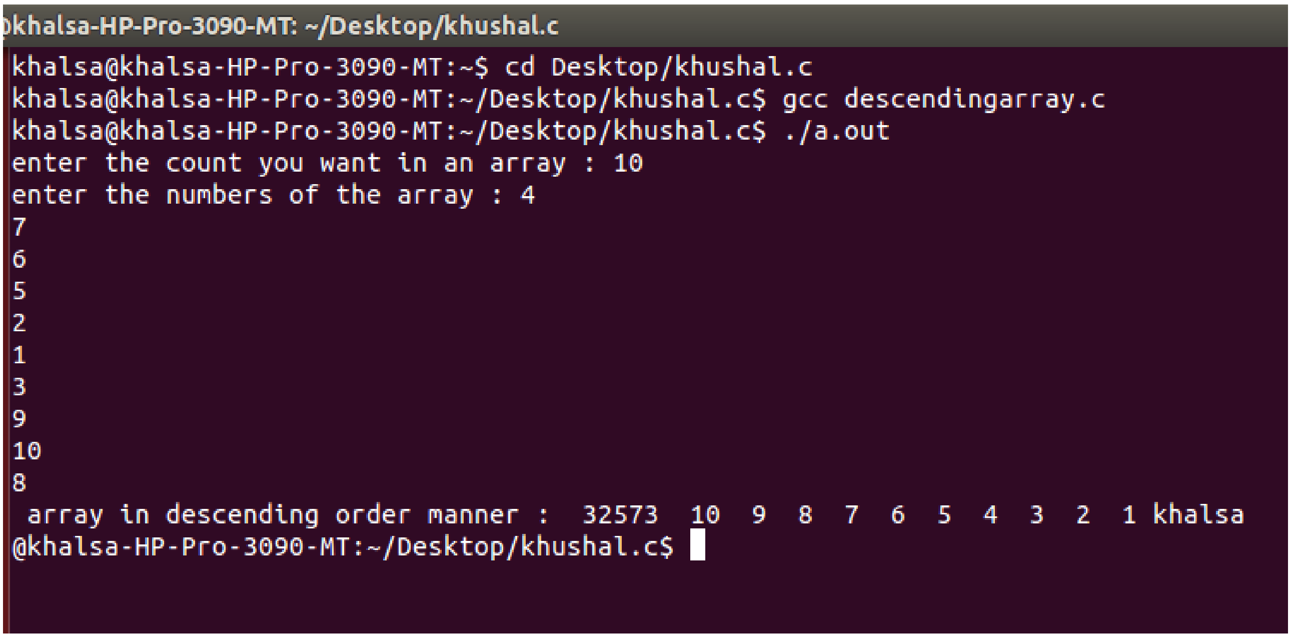
FLOW CHART ::


START
ENTER COUNT
FOR (I<N)
ENTER A[I]
FOR (I<N)
FOR (J<N)
if(A[J]>A[J+1])
b=A[J]
A[J]=A[J+1]
A[J+1]=b
END

ARRAY IN DESCENDING ORDER

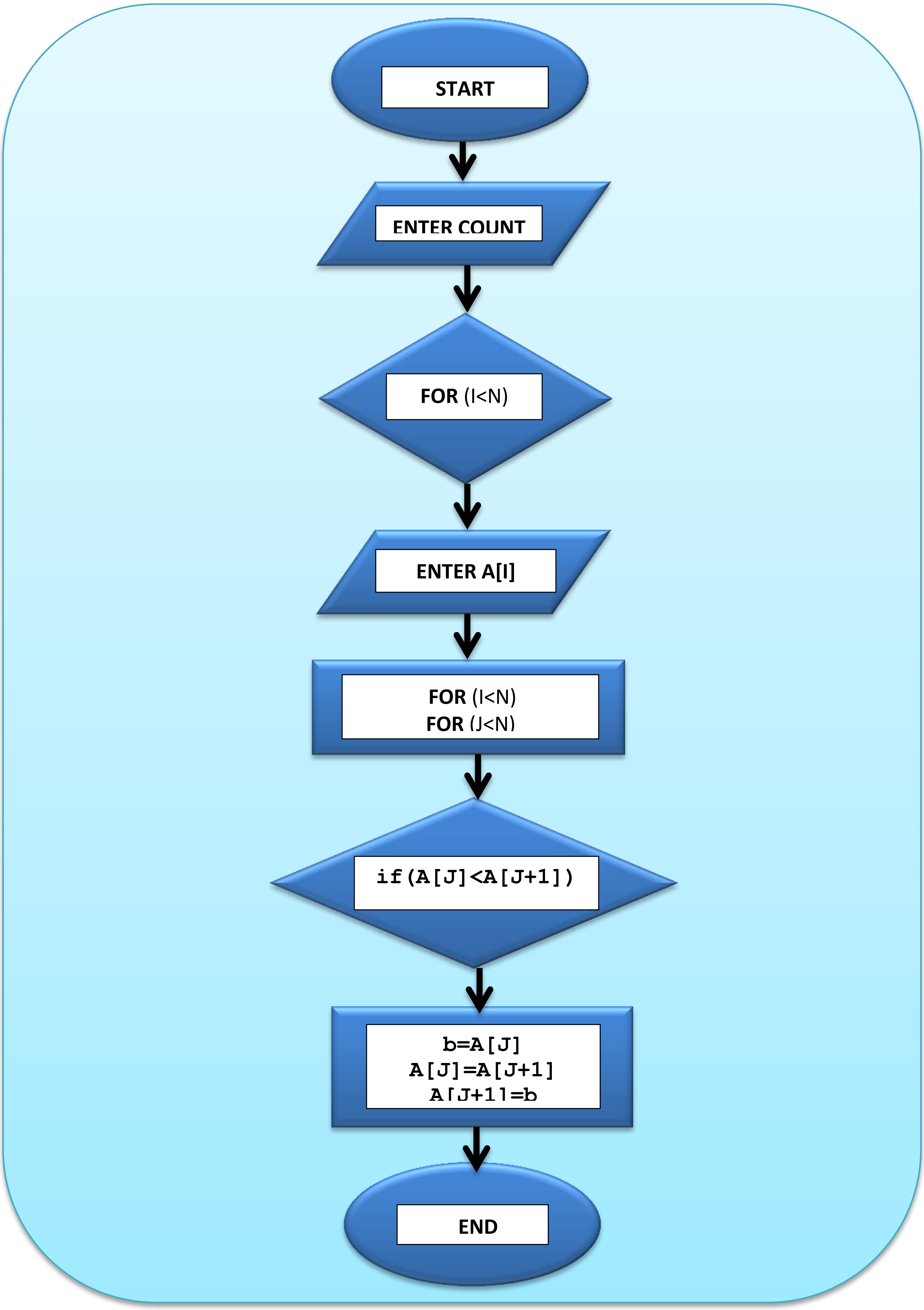
```
#include<stdio.h>
void main()
{
      int a[100],n,i,j,b;
      printf("enter the count you want in an array : ");
      scanf("%d",&n);
      printf("enter the numbers of the array : ");
      for(i=1;i<=n;i++)
      {
            scanf("%d",&a[i]);
      }
      for(i=1;i<=n;i++)
      {
            for(j=1;j<=n;j++)
            {
                  if(a[j]>a[j+1])
                  {
                        b=a[j];
                        a[j]=a[j+1];
                        a[j+1]=b;
                  }
            }
      }
      printf(" array in descending order manner : ");
      for(i=(n+1);i>=1;i--)
      {
            printf(" %d ",a[i]);
      }

}
```

OUTPUT ::



```
)khalsa-HP-Pro-3090-MT: ~/Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc descendingarray.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
enter the count you want in an array : 10
enter the numbers of the array : 4
7
6
5
2
1
3
9
10
8
 array in descending order manner :  32573  10  9  8  7  6  5  4  3  2  1 khalsa
@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$
```

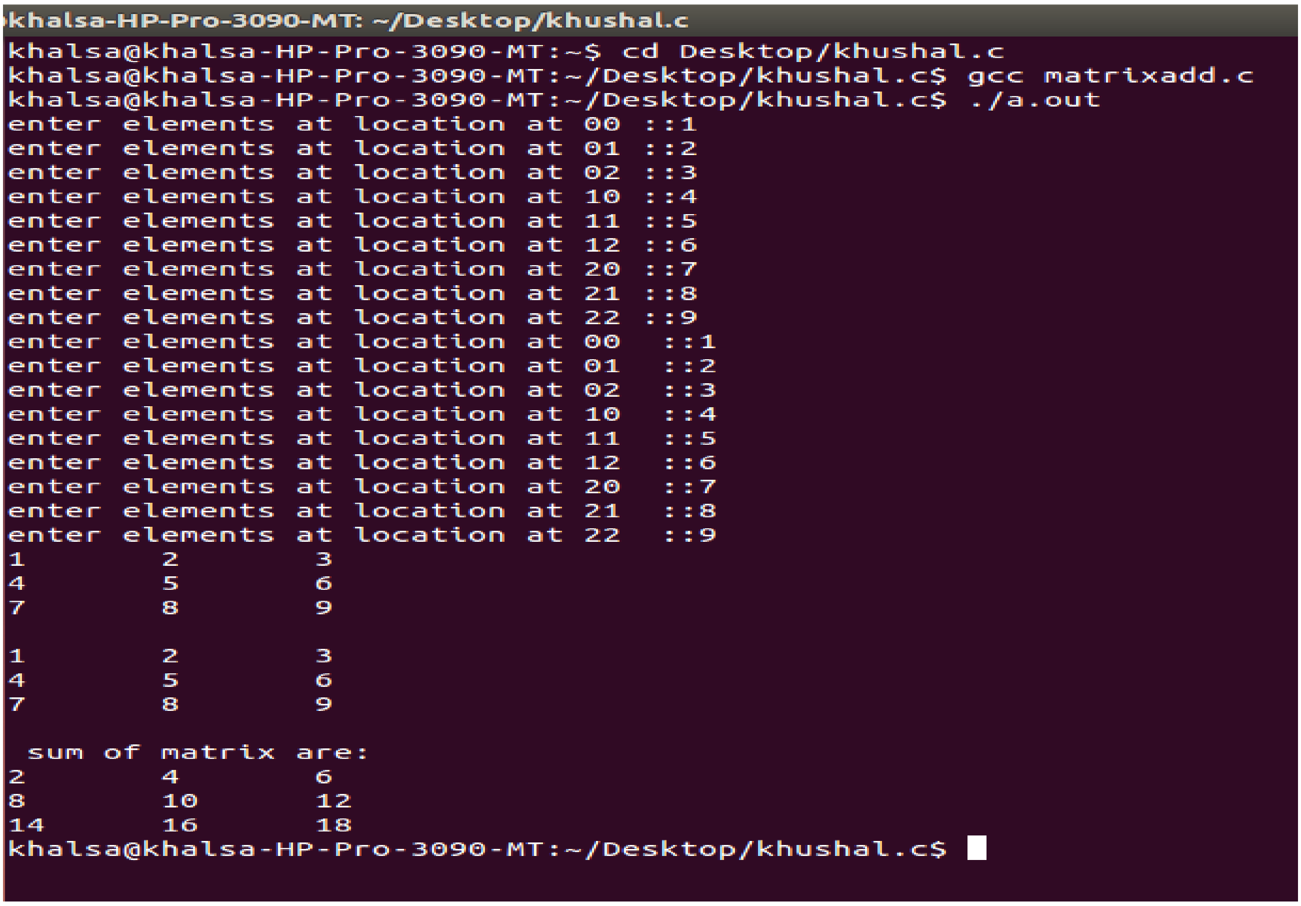
FLOW CHART ::


START
ENTER COUNT
FOR (I<N)
ENTER A[I]
FOR (I<N)
FOR (J<N)
if(A[J]<A[J+1])
b=A[J]
A[J]=A[J+1]
A[J+1]=b
END

SUM OF TWO MATRICES

```
#include<stdio.h>
void main()
{
      int A[3][3],B[3][3],i,j;
      int C[3][3];
      for(i=0;i<3;i++)
      for(j=0;j<3;j++)
{
      printf("enter elements at location at %d%d ::",i,j);
      scanf("%d",&A[i][j]);
}
      for(i=0;i<3;i++)
      for(j=0;j<3;j++)
{
      printf("enter elements at location at %d%d  ::",i,j);
      scanf("%d",&B[i][j]);
}
      for(i=0;i<3;i++)
{
      for(j=0;j<3;j++)
{
      printf("%d ",A[i][j]);
}
      printf("\n");
}
      printf("\n");
      for(i=0;i<3;i++)
{
      for(j=0;j<3;j++)
{
      printf("%d ",B[i][j]);
}
      printf("\n");
}
      for(i=0;i<3;i++)
      for(j=0;j<3;j++)
{
      C[i][j]=A[i][j]+B[i][j];
}
      printf("\n sum of matrix are: \n");
      for(i=0;i<3;i++)
{
      for(j=0;j<3;j++)
{
      printf("%d ",C[i][j]);
}
      printf("\n");
}
}
```

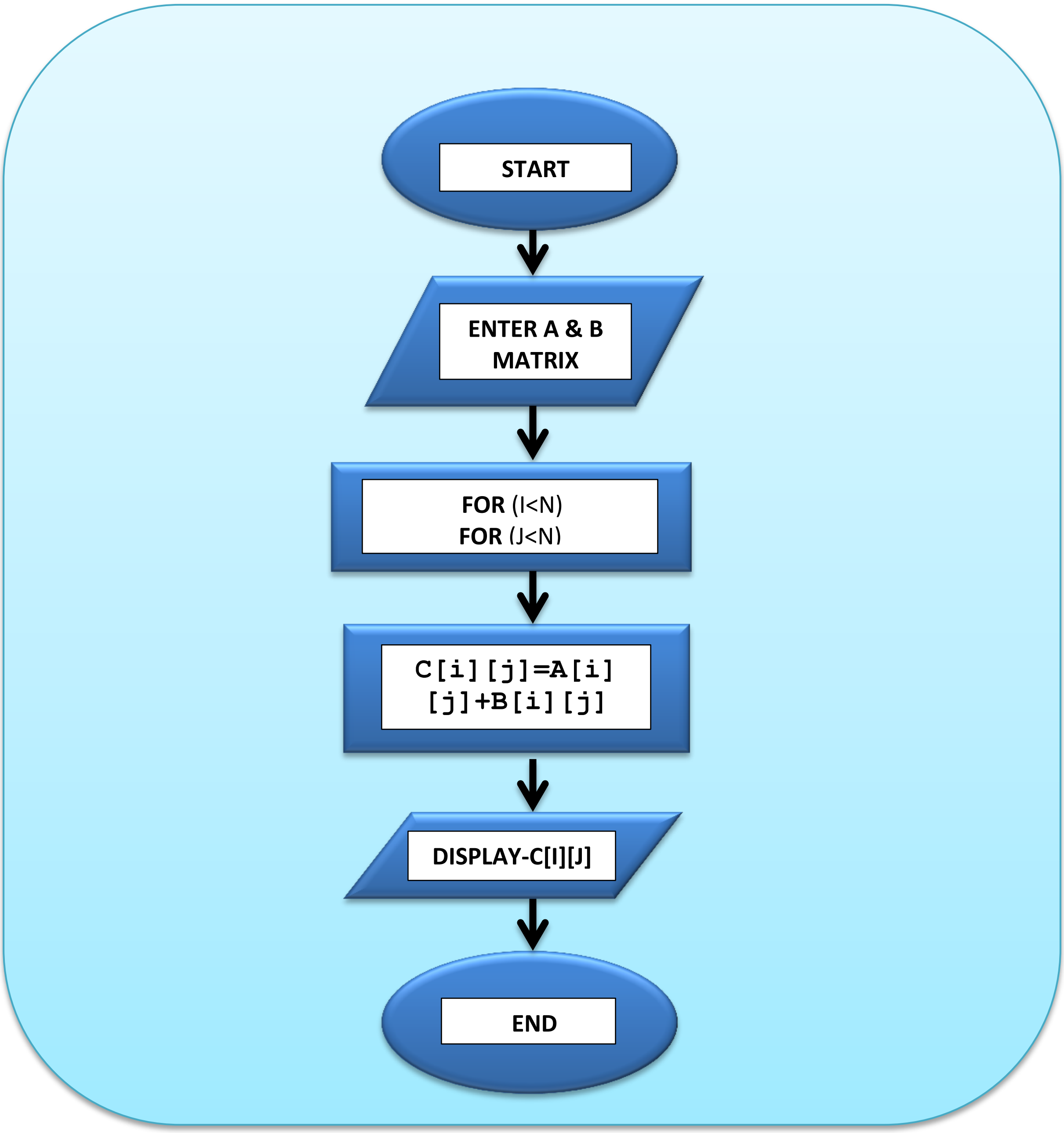
OUTPUT ::



khalsa-HP-Pro-3090-MT: ~/Desktop/khushal.c

```
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc matrixadd.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
enter elements at location at 00 ::1
enter elements at location at 01 ::2
enter elements at location at 02 ::3
enter elements at location at 10 ::4
enter elements at location at 11 ::5
enter elements at location at 12 ::6
enter elements at location at 20 ::7
enter elements at location at 21 ::8
enter elements at location at 22 ::9
enter elements at location at 00  ::1
enter elements at location at 01  ::2
enter elements at location at 02  ::3
enter elements at location at 10  ::4
enter elements at location at 11  ::5
enter elements at location at 12  ::6
enter elements at location at 20  ::7
enter elements at location at 21  ::8
enter elements at location at 22  ::9
1       2       3
4       5       6
7       8       9

1       2       3
4       5       6
7       8       9

 sum of matrix are:
2       4       6
8       10      12
14      16      18
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ 
```

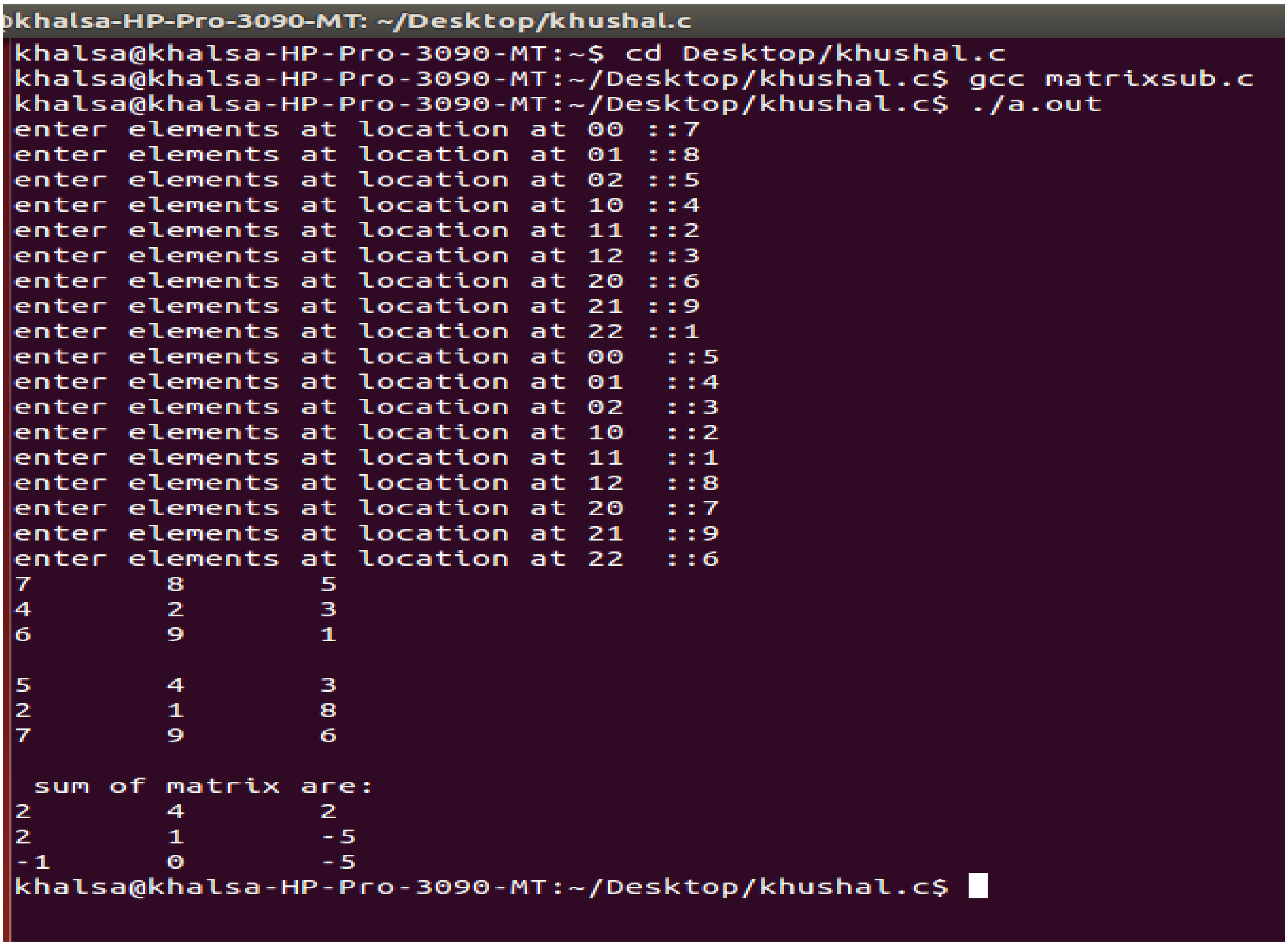
FLOW CHART ::


START
ENTER A & B MATRIX
FOR (I<N)
FOR (J<N)
C[i][j]=A[i][j]+B[i][j]
DISPLAY-C[I][J]
END

DIFFERENCE BETWEEN TWO MATRICES

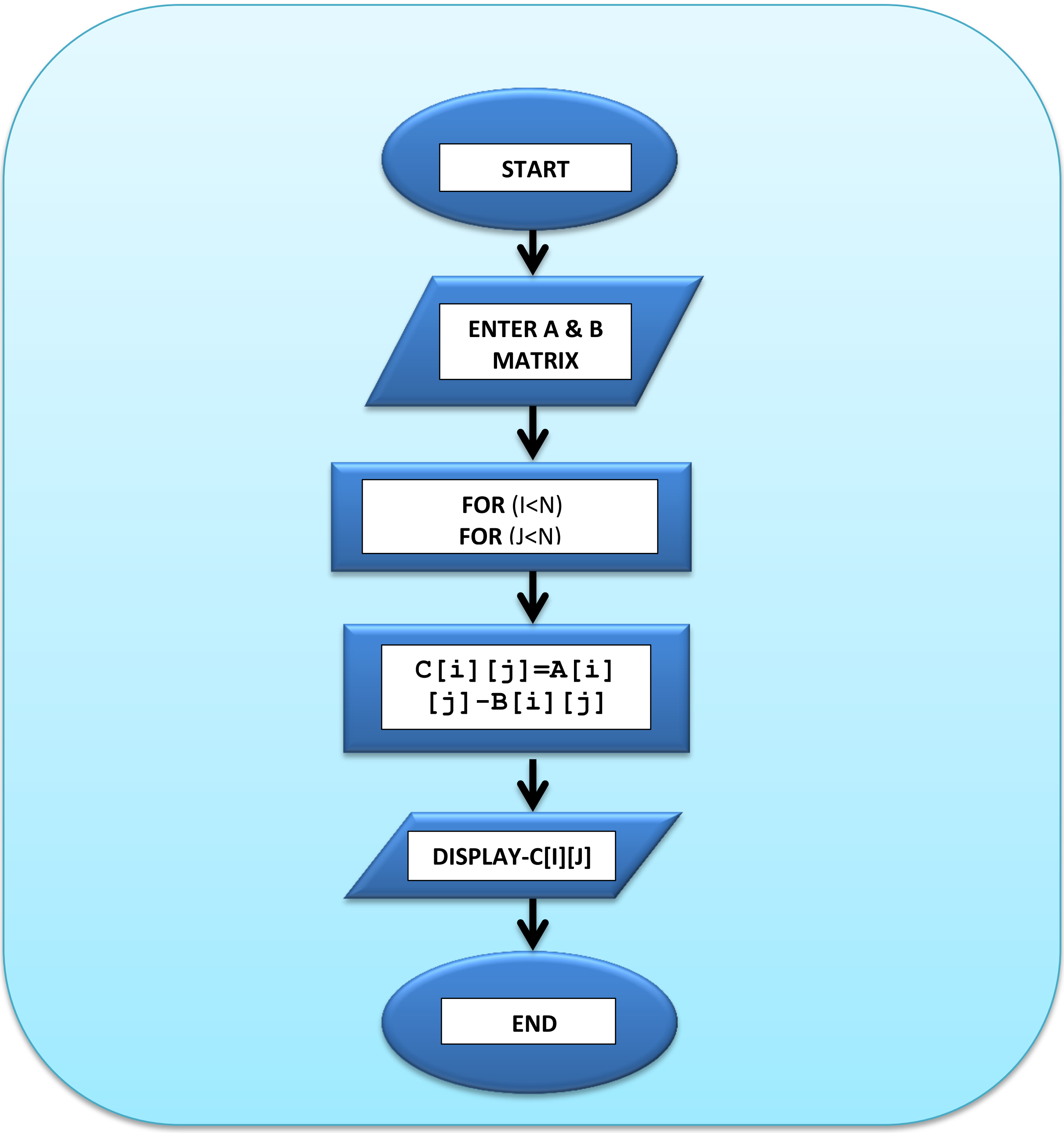
```
#include<stdio.h>
void main()
{
      int A[3][3],B[3][3],i,j;
      int C[3][3];
      for(i=0;i<3;i++)
      for(j=0;j<3;j++)
{
      printf("enter elements at location at %d%d ::",i,j);
      scanf("%d",&A[i][j]);
}
      for(i=0;i<3;i++)
      for(j=0;j<3;j++)
{
      printf("enter elements at location at %d%d  ::",i,j);
      scanf("%d",&B[i][j]);
}
      for(i=0;i<3;i++)
{
      for(j=0;j<3;j++)
{
      printf("%d ",A[i][j]);
}
      printf("\n");
}
      printf("\n");
      for(i=0;i<3;i++)
{
      for(j=0;j<3;j++)
{
      printf("%d ",B[i][j]);
}
      printf("\n");
}
      for(i=0;i<3;i++)
      for(j=0;j<3;j++)
{
      C[i][j]=A[i][j]-B[i][j];
}
      printf("\n difference of matrix are: \n");
      for(i=0;i<3;i++)
{
      for(j=0;j<3;j++)
{
      printf("%d ",C[i][j]);
}
      printf("\n");
}
}
```

OUTPUT ::



```
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc matrixsub.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
enter elements at location at 00 ::7
enter elements at location at 01 ::8
enter elements at location at 02 ::5
enter elements at location at 10 ::4
enter elements at location at 11 ::2
enter elements at location at 12 ::3
enter elements at location at 20 ::6
enter elements at location at 21 ::9
enter elements at location at 22 ::1
enter elements at location at 00  ::5
enter elements at location at 01  ::4
enter elements at location at 02  ::3
enter elements at location at 10  ::2
enter elements at location at 11  ::1
enter elements at location at 12  ::8
enter elements at location at 20  ::7
enter elements at location at 21  ::9
enter elements at location at 22  ::6
7       8       5
4       2       3
6       9       1

5       4       3
2       1       8
7       9       6

 sum of matrix are:
2       4       2
2       1       -5
-1      0       -5
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ 
```

FLOW CHART ::

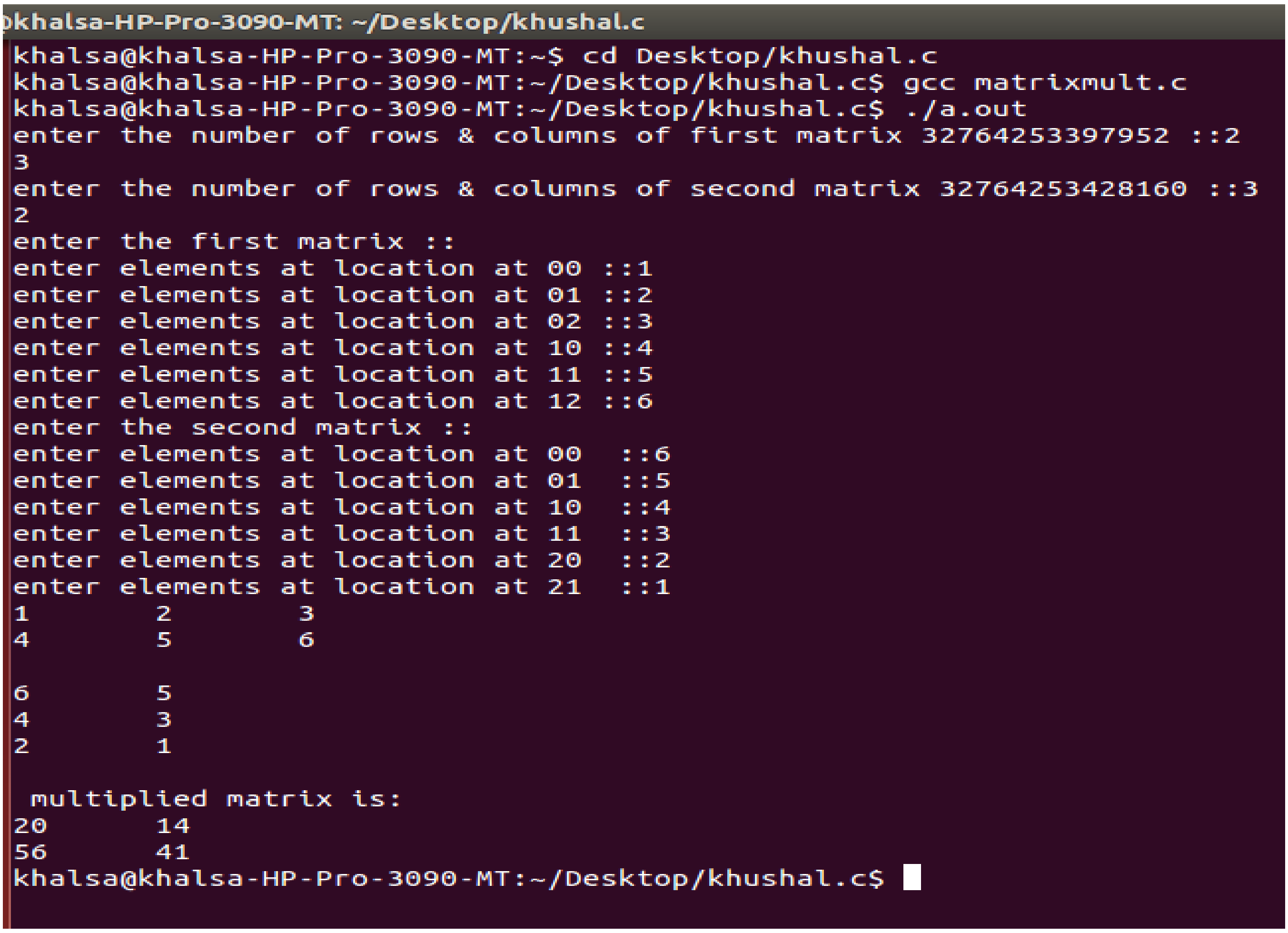

START
ENTER A & B MATRIX
FOR (I<N)
FOR (J<N)
C[i][j]=A[i][j]-B[i][j]
DISPLAY-C[I][J]
END

PRODUCT OF TWO MATRICES

```
#include<stdio.h>
void main()
{
     int A[10][10],B[10][10],i,j,m,n,p,q,k;
     int C[10][10];
     printf("enter the number of rows & columns of first matrix
%d%d ::",m,n);
     scanf("%d%d",&m,&n);
     printf("enter the number of rows & columns of second matrix
%d%d ::",p,q);
     scanf("%d%d",&p,&q);
     printf("enter the first matrix ::\n");
     for(i=0;i<m;i++)
     for(j=0;j<n;j++)
     {
          printf("enter elements at location at %d%d ::",i,j);
          scanf("%d",&A[i][j]);
     }
     printf("enter the second matrix ::\n");
     for(i=0;i<p;i++)
     for(j=0;j<q;j++)
     {
          printf("enter elements at location at %d%d  ::",i,j);
          scanf("%d",&B[i][j]);
     }
     for(i=0;i<m;i++)
     {
          for(j=0;j<n;j++)
          {
               printf("%d ",A[i][j]);
          }
          printf("\n");
     }
     printf("\n");
     for(i=0;i<p;i++)
     {
          for(j=0;j<q;j++)
          {
               printf("%d ",B[i][j]);
          }
          printf("\n");
     }
     if(n==p)
     {

          for(i=0;i<m;i++)
          {
          for(j=0;j<q;j++)
          {
               C[i][j]=0;
               for(k=0;k<n;k++)
               C[i][j]=C[i][j]+(A[i][k]*B[k][j]);
          }
          }
          printf("\n multiplied matrix is: \n");
```

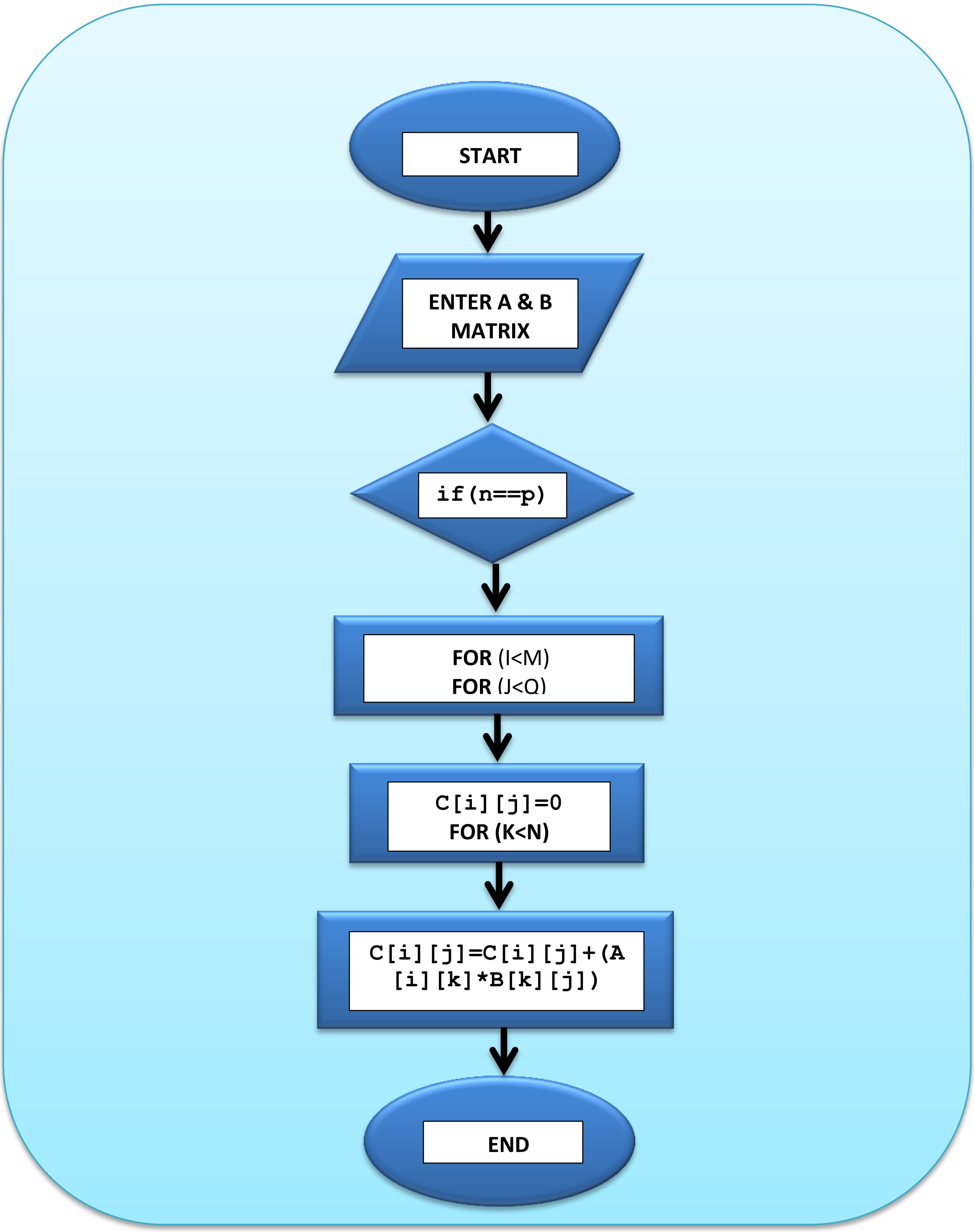
```
            for(i=0;i<m;i++)
            {
                for(j=0;j<q;j++)
                {
                    printf("%d ",C[i][j]);
                }
                printf("\n");
            }
        }
        else
        printf("\n matrix can't be multiplied \n");
    }
```

OUTPUT ::



```
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc matrixmult.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
enter the number of rows & columns of first matrix 32764253397952 ::2
3
enter the number of rows & columns of second matrix 32764253428160 ::3
2
enter the first matrix ::
enter elements at location at 00 ::1
enter elements at location at 01 ::2
enter elements at location at 02 ::3
enter elements at location at 10 ::4
enter elements at location at 11 ::5
enter elements at location at 12 ::6
enter the second matrix ::
enter elements at location at 00  ::6
enter elements at location at 01  ::5
enter elements at location at 10  ::4
enter elements at location at 11  ::3
enter elements at location at 20  ::2
enter elements at location at 21  ::1
1       2       3
4       5       6

6       5
4       3
2       1

 multiplied matrix is:
20      14
56      41
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ █
```

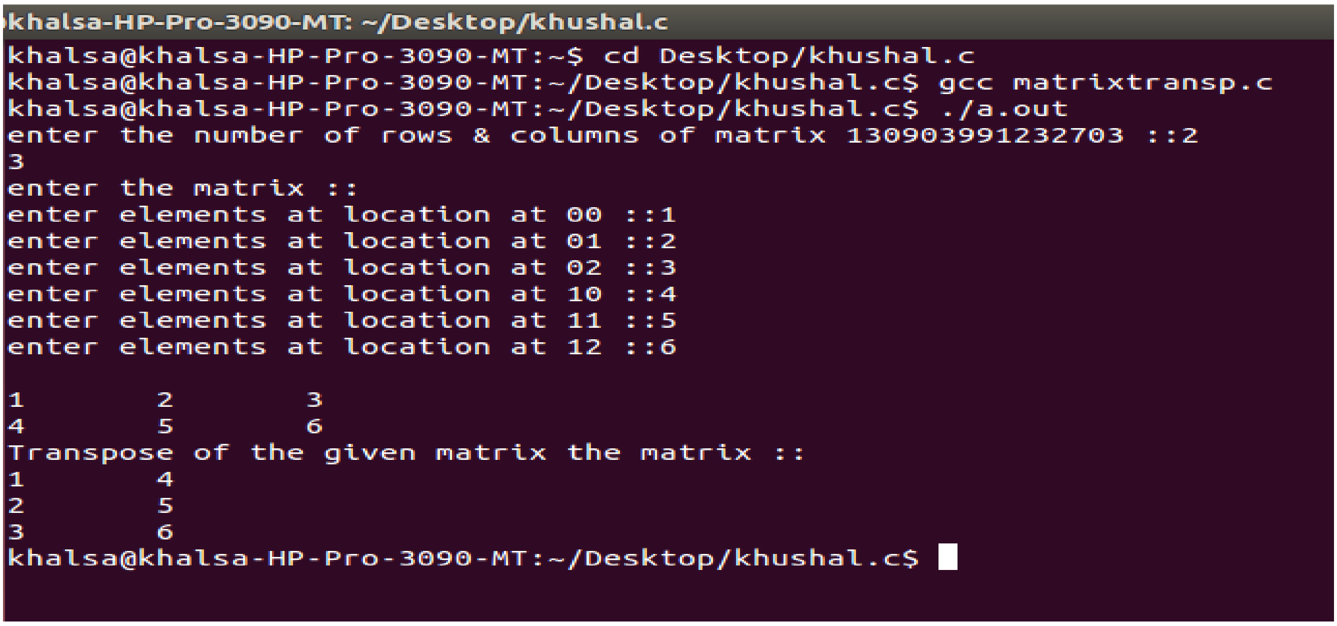
FLOW CHART ::


START
ENTER A & B MATRIX
if(n==p)
FOR (I<M)
FOR (J<Q)
C[i][j]=0
FOR (K<N)
C[i][j]=C[i][j]+(A
[i][k]*B[k][j])
END

TRANSPOSE OF A MATRIX

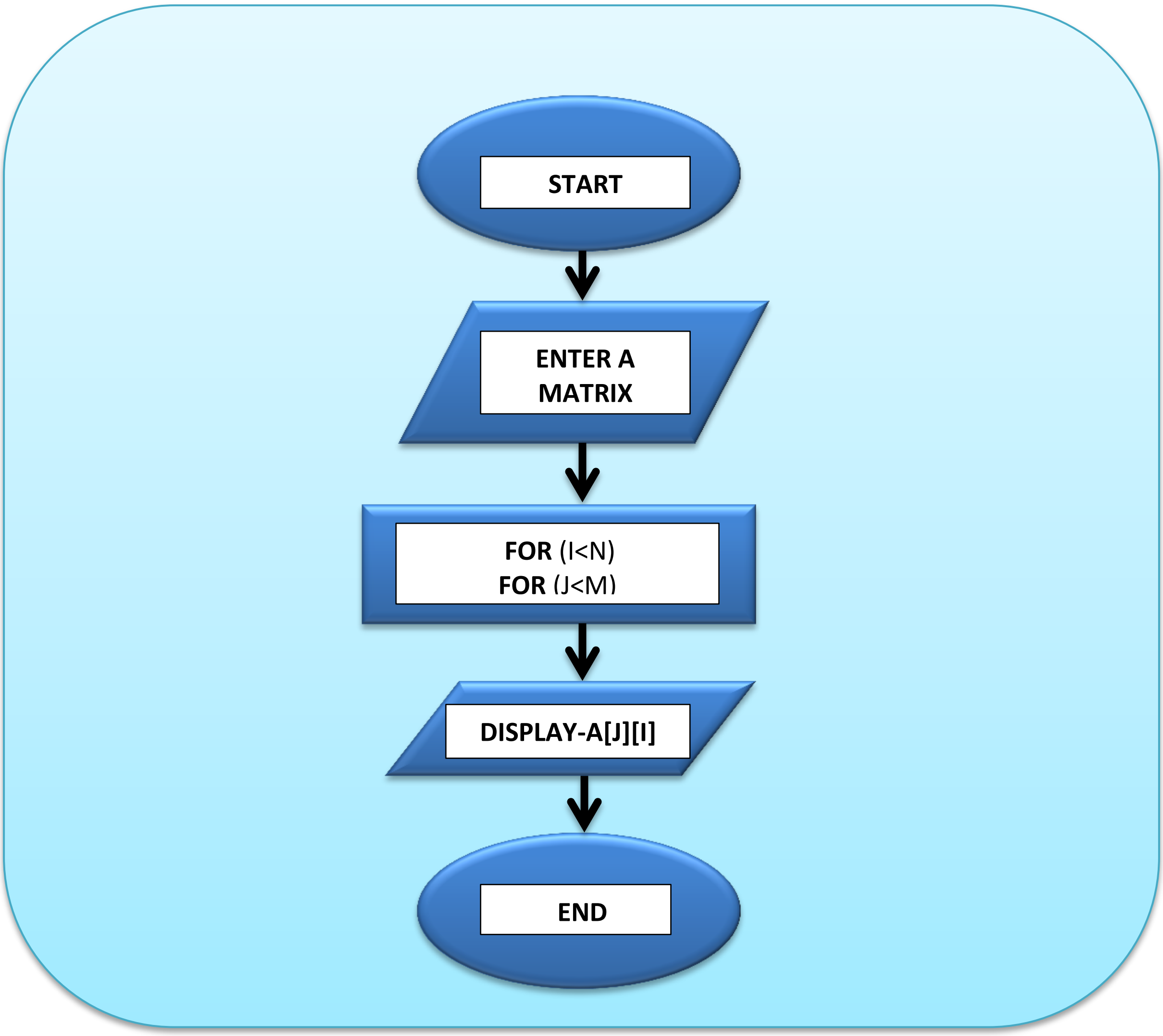
```
#include<stdio.h>
void main()
{
      int A[10][10],i,j,m,n;
      printf("enter the number of rows & columns of matrix %d%d
::",m,n);
      scanf("%d%d",&m,&n);
      printf("enter the matrix ::\n");
      for(i=0;i<m;i++)
      for(j=0;j<n;j++)
      {
            printf("enter elements at location at %d%d ::",i,j);
            scanf("%d",&A[i][j]);
      }
      printf("\n");
      for(i=0;i<m;i++)
      {
            for(j=0;j<n;j++)
            {
                  printf("%d ",A[i][j]);
            }
            printf("\n");
      }
      printf("Transpose of the given matrix the matrix ::\n");
      for(i=0;i<n;i++)
      {
            for(j=0;j<m;j++)
            {
                  printf("%d ",A[j][i]);
            }
            printf("\n");
      }
}
```

OUTPUT ::



```
khalsa-HP-Pro-3090-MT: ~/Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc matrixtransp.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
enter the number of rows & columns of matrix 130903991232703 ::2
3
enter the matrix ::
enter elements at location at 00 ::1
enter elements at location at 01 ::2
enter elements at location at 02 ::3
enter elements at location at 10 ::4
enter elements at location at 11 ::5
enter elements at location at 12 ::6

1       2       3
4       5       6
Transpose of the given matrix the matrix ::
1       4
2       5
3       6
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ 
```

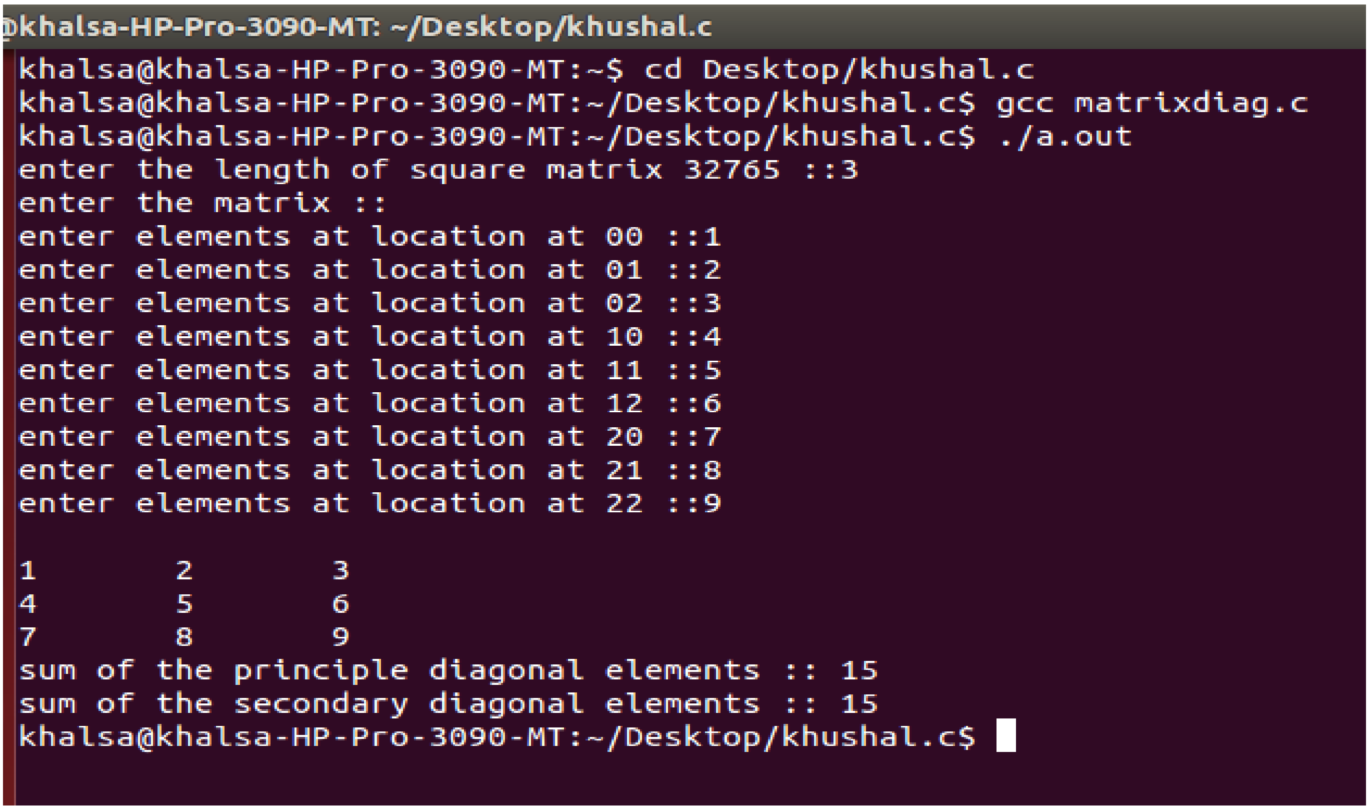
FLOW CHART ::



SUM OF PRINCIPLE & SECONDARY DIAGONAL ELEMENTS

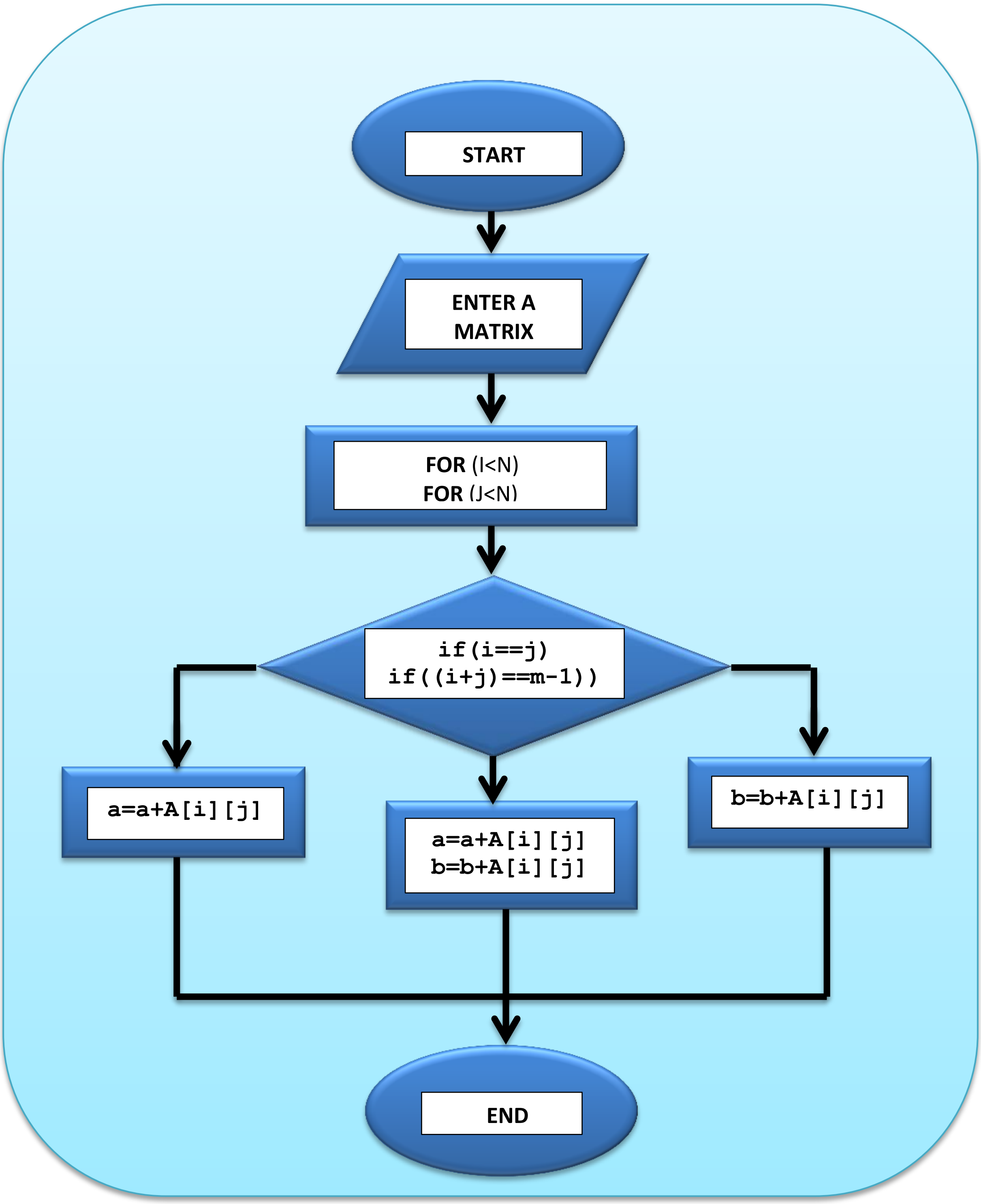
```
#include<stdio.h>
void main()
{
      int A[10][10],i,j,m,n,a=0,b=0;
      printf("enter the length of square matrix %d ::",m);
      scanf("%d",&m);
      printf("enter the matrix ::\n");
      for(i=0;i<m;i++)
      for(j=0;j<m;j++)
      {
            printf("enter elements at location at %d%d ::",i,j);
            scanf("%d",&A[i][j]);
      }
      printf("\n");
      for(i=0;i<m;i++)
      {
            for(j=0;j<m;j++)
            {
                  printf("%d ",A[i][j]);
            }
            printf("\n");
      }
      for(i=0;i<m;i++)
      {
            for(j=0;j<m;j++)
            {
            if((i==j)&&((i+j)==(m-1)))
            {
                  a=a+A[i][j];
                  b=b+A[i][j];
            }
            else if(i==j)
                  a=a+A[i][j];
            else if((i+j)==(m-1))
                  b=b+A[i][j];
            }
      }
      printf("sum of the principle diagonal elements :: %d\n",a);
      printf("sum of the secondary diagonal elements :: %d\n",b);
}
```

OUTPUT ::



```
@khalsa-HP-Pro-3090-MT: ~/Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc matrixdiag.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
enter the length of square matrix 32765 ::3
enter the matrix ::
enter elements at location at 00 ::1
enter elements at location at 01 ::2
enter elements at location at 02 ::3
enter elements at location at 10 ::4
enter elements at location at 11 ::5
enter elements at location at 12 ::6
enter elements at location at 20 ::7
enter elements at location at 21 ::8
enter elements at location at 22 ::9

1       2       3
4       5       6
7       8       9
sum of the principle diagonal elements :: 15
sum of the secondary diagonal elements :: 15
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ 
```

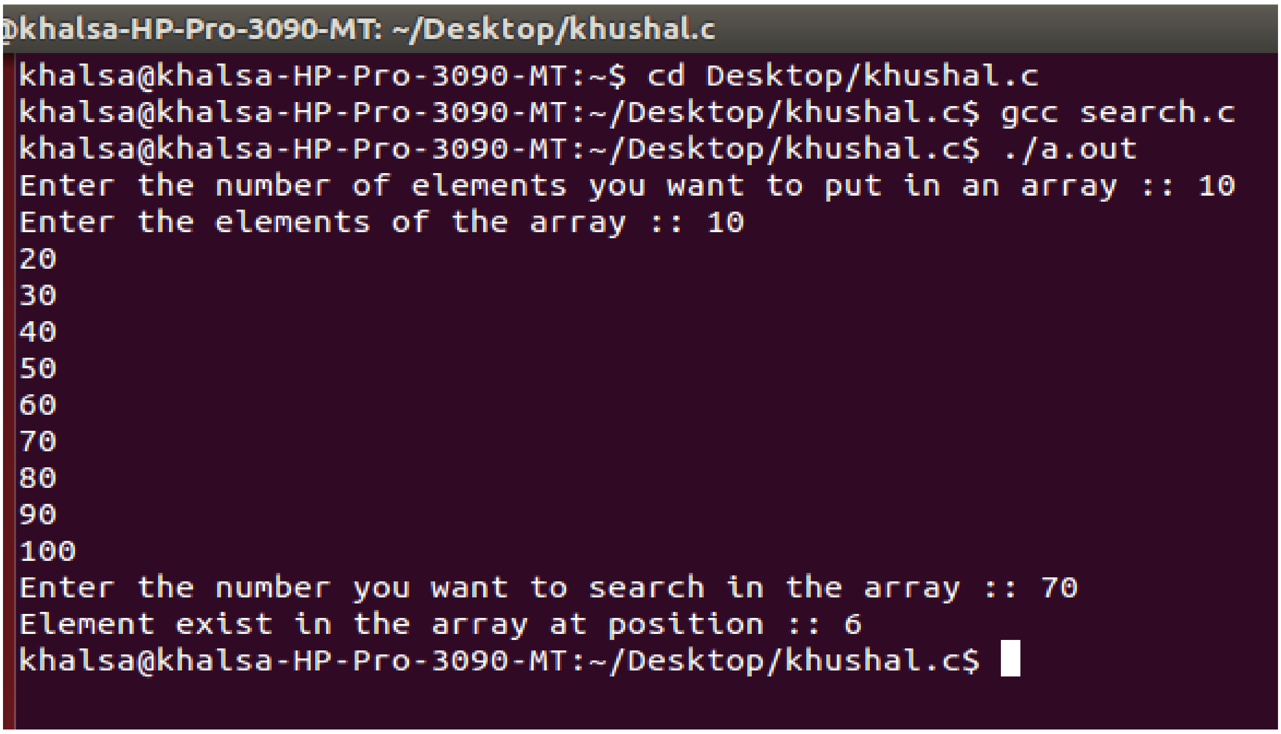
FLOW CHART ::


START
ENTER A MATRIX
FOR (I<N)
FOR (J<N)
if(i==j)
if((i+j)==m-1))
a=a+A[i][j]
a=a+A[i][j]
b=b+A[i][j]
b=b+A[i][j]
END

LINEAR SEARCH

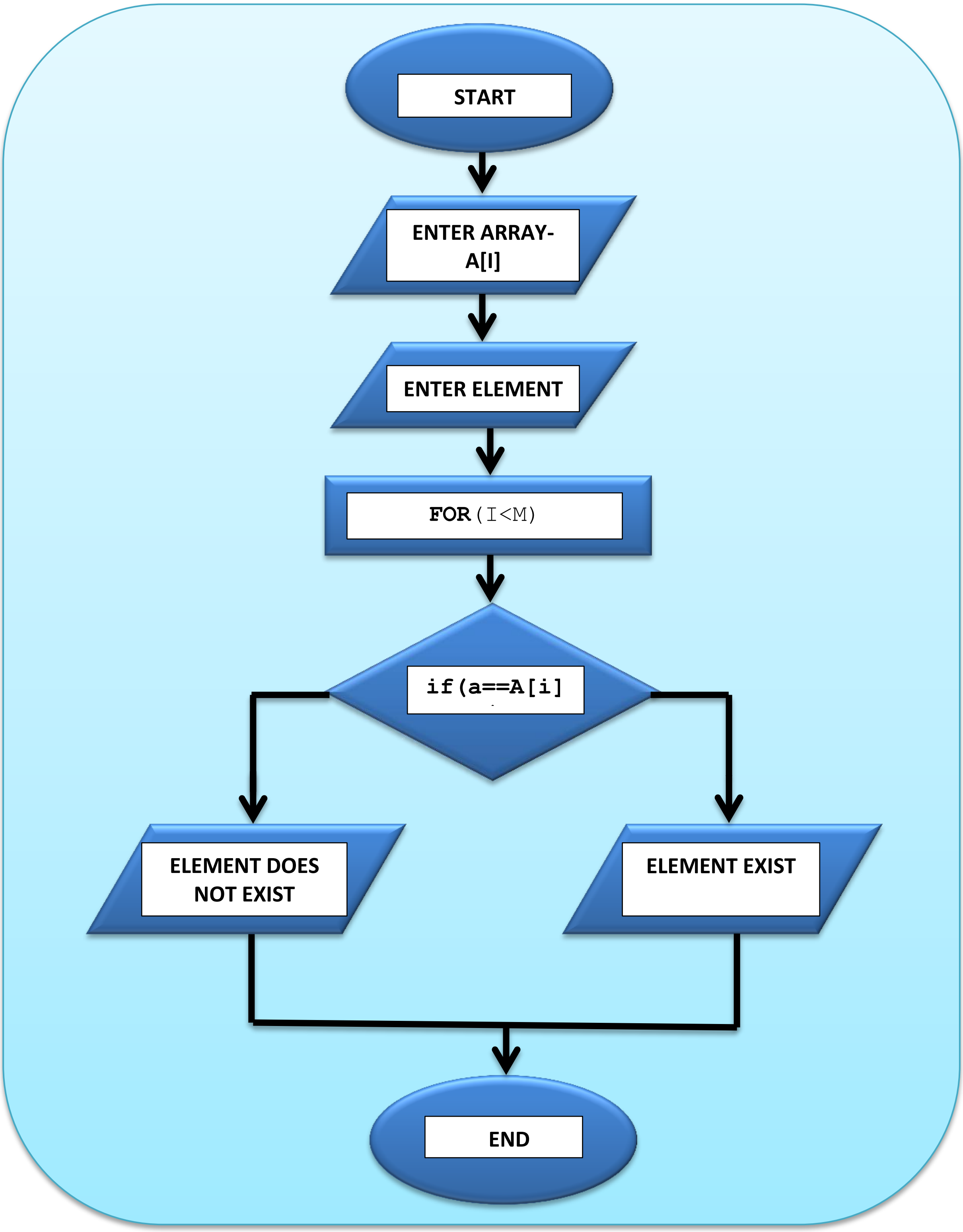
```
#include<stdio.h>
void main()
{
      int A[100],m,i,a,f=0;
      printf("Enter the number of elements you want to put in an
array :: ");
      scanf("%d",&m);
      printf("Enter the elements of the array :: ");
      for(i=0;i<m;i++)
            scanf("%d",&A[i]);
      printf("Enter the number you want to search in the array ::
");
      scanf("%d",&a);
      for(i=0;i<m;i++)
      {
            if(a==A[i])
            {
                  f=1;
                  break;
            }
            else
                  f=0;
      }
      if(f==1)
            printf("Element exist in the array at position :: %d
\n",i);
      else
            printf("Element does not exist in the array :: ");
}
```

OUTPUT ::



```
khalsa@khalsa-HP-Pro-3090-MT:~$ cd Desktop/khushal.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ gcc search.c
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$ ./a.out
Enter the number of elements you want to put in an array :: 10
Enter the elements of the array :: 10
20
30
40
50
60
70
80
90
100
Enter the number you want to search in the array :: 70
Element exist in the array at position :: 6
khalsa@khalsa-HP-Pro-3090-MT:~/Desktop/khushal.c$
```

FLOW CHART ::


START
ENTER ARRAY-A[I]
ENTER ELEMENT
FOR(I<M)
if(a==A[i]
ELEMENT DOES NOT EXIST
ELEMENT EXIST
END