SOFTWARE DEVELOPMENT LAB I (C PROGRAMMING)

(Course Code: MCACP107)

Submitted In Partial Fulfilment Of The Requirement For The Award Of The Degree Of

MASTER OF COMPUTER APPLICATION

SEMESTER 1 [2024-2026]

Submitted By

ARUN SURESH

Register No: 243242210882

Under the guidance of

Dr. Susheel George Joseph

DEPARTMENT OF COMPUTER APPLICATION,

**KRISTU JYOTI COLLEGE OF MANAGEMENT AND TECHNOLOGY,
CHANGANASSERY**

January-2025

KRISTU JYOTI COLLEGE OF MANAGEMENT AND TECHNOLOGY CHANGANACHERRY

DEPARTMENT OF COMPUTER APPLICATION

CERTIFICATE

Certify that the practical laboratory record of Software Development Lab I (C Programming) is a Bonafide report of the practical works done by ARUN SURESH (Reg No: 243242210882) under the guidance and supervision is submitted in partial fulfilment of Master of Computer Applications, awarded by Mahatma Gandhi University, Kerala.

Dr. Susheel George Joseph……………………………………..

(Teacher-in-Charge)

Mr. Roji Thomas………………………………………………

(H.O.D)

Rev.Fr. Joshy Cheeramkuzhy CMI…………………………..

(Principal)

Submitted for practical examination and viva held on……………………

Department Seal

Examiner(s)

1.

2.

DECLARATION

I hereby declare that the project work entitled "**SOFTWARE LAB I**" submitted to Mahatma Gandhi University in partial fulfilment of requirement for the award of post-graduation of Master of Computer Application from Kristu Jyoti College of management and technology, Changanacherry is a record of bonafide work done under the guidance of Dr. Susheel George Joseph, Department of Computer Application. This project work has not been submitted in partial or fulfilment of any other post-graduation or similar of this University or any other university.

ARUN SURESH

Reg No: 243242210882

ACKNOWLEDGEMENT

I would like to express my thankfulness towards a number of people who have been instrument in making of this project. First of all, we thank the college management who has been with us with most helpful facilities throughout the completion of the project. I am greatly indebted to **Rev. Fr. Joshy Cheeramkuzhy CMI**, Principal of Kristu Jyoti College who whole heartedly give us the permission and whose advice was a real encouragement. I record my sincere thanks and gratitude to **Mr. Roji Thomas**, HOD, Computer Application, for the valuable suggestions provided. I am deeply indebted to Asst. Prof. **Dr. Susheel George Joseph** Department of Computer Application, for the valuable discussion and helpful counselling. Last, not the least I wish to express my gratitude and heartfelt thanks to our friends, family and whom with their valuable advice, encouragement and support for the successful completion of the project.

Above all thanking the GOD the Almighty.

ARUN SURESH

CONTENTS

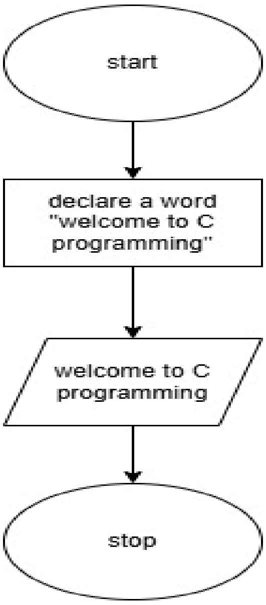
Basic programs

1. Write a C program to print a word

Algorithm

```
Step 1: Start
Step 2: Declare  a word "welcome to C programming"
Step 3: Print welcome to C programming
Step 4: Stop
```

Flowchart



Program

```
 #include<stdio.h>
#include<conio.h>
void main
{
printf("welcome to C programming\n");
 getch();
}
```

Output

welcome to C programming

2. Write a C program to Declare Variable and Printing Its Value

Algorithm

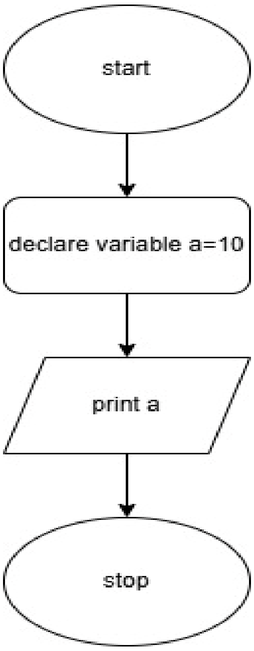
Step 1:Start

Step 2: Declare an integer variable a and initialize it with a value (e.g., a = 10).

Step 3: Use the printf function to display the value of a.

Step 4: End the program.

Flowchart



Program

```
#include <stdio.h>
#include <coni.h>
void main()
{
 int a = 10;
  printf("The value of a is: %d\n", a);
   return 0;
}
```

Output

```
The value of a is: 10
```

3. Write a C program to calculate area of rectangle

Algorithm

Step 1: Start

Step 2: Declare variables width, height, area

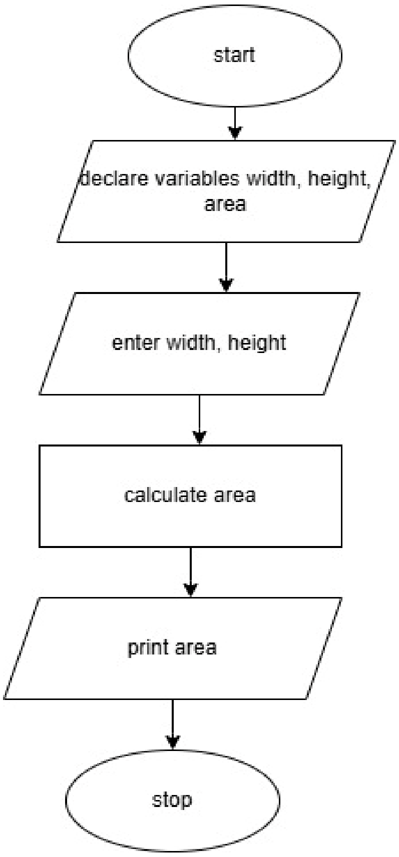
Step 3: Read width, height

Step 4: Calculate area

Step 5: Print area

Step 6: Stop

Flowchart



Program

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

```
#include<conio.h>
 void main ()
{
int height, width; float area;
clrscr ();
printf (“Enter width:”);
scanf (“%d”, &width);
printf (“Enter height:”);
scanf (“%d”, &height);
area=width*height;
printf (“\n\n Area of rectangle: %f ”,area);
 getch ();
}
```

Output

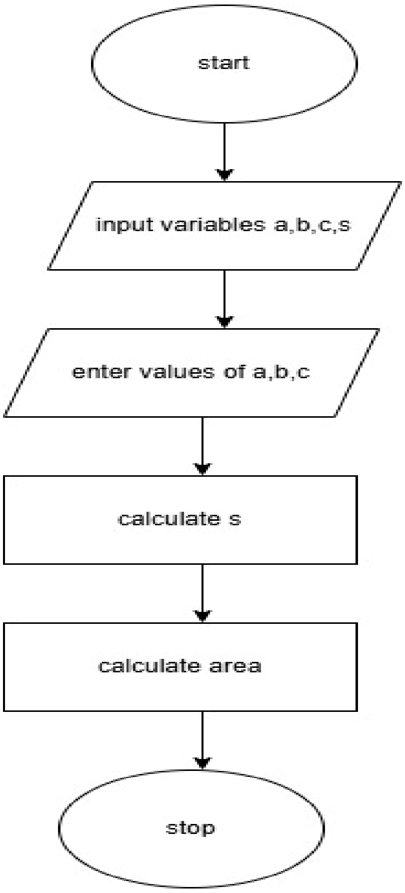
```
Enter width:3
 Enter height:2
Area of rectangle:6.000
```

4. Write a C program to calculate area of scalene triangle

Algorithm

```
Step 1: Start
Step 2: Input variable a, b, c, s, area
 Step 3: Read values of a, b, c
Step 4: Calculate s
Step 5: Calculate area
Step 6: Print area
 Step 7: Stop
```

Flowchart



Program

```
#include<stdio.h>
 #include<conio.h>
#include<math.h>
void main ()
{
float a, b, c, s, area;
 clrscr ();
printf (“Enter the value of a, b and c: ”);
 scanf(“%d%d%d”, &a,&b,&c);
s=(a+b+c)/2;
area=sqrt(s*(s-a)*(s-b)*(s-c));
 printf(“\n\n Area of a triangle:%d”,area); getch(); }
```

Output

```
Enter the value of a, b and c: 3 4 5
 Area of a triangle: 6.000000
```

5. Write a C program to calculate Area and Circumference of Circle

Algorithm

Step 1: Start

Step 2: Declare variables area, radius and circumference

Step 3: Read variable radius

Step 4: Print value radius

Step 5: Calculate area

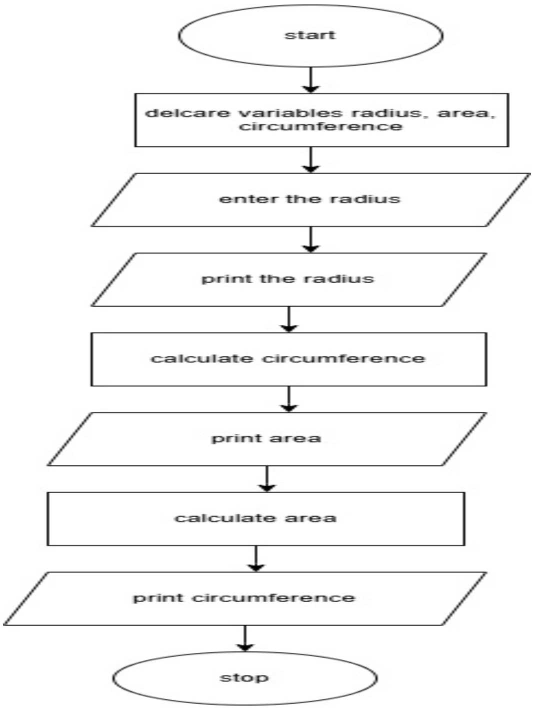
Step 6: Calculate circumference

Step 7: Print value area

Step 8: Print value circumference

Step 9: Stop

Flowchart



Program

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

Output

```
Enter the radius of a circle: 3
 Radius= 3
Area= 28.26
Circumference= 18.84
```

6. write a C program to check whether number is prime or not

Algorithm

```
Step 1: Start
Step 2: Declare variables n, i, flag
Step 3: Initialize variables flag=1,i=2
Step 4: Read n
Step 5: if (n<=1)
    Print ‘n is not a prime number’ Go to step 7
```

Step 6: Repeat the step until i< [(n/2)+1] if(n%i==0)

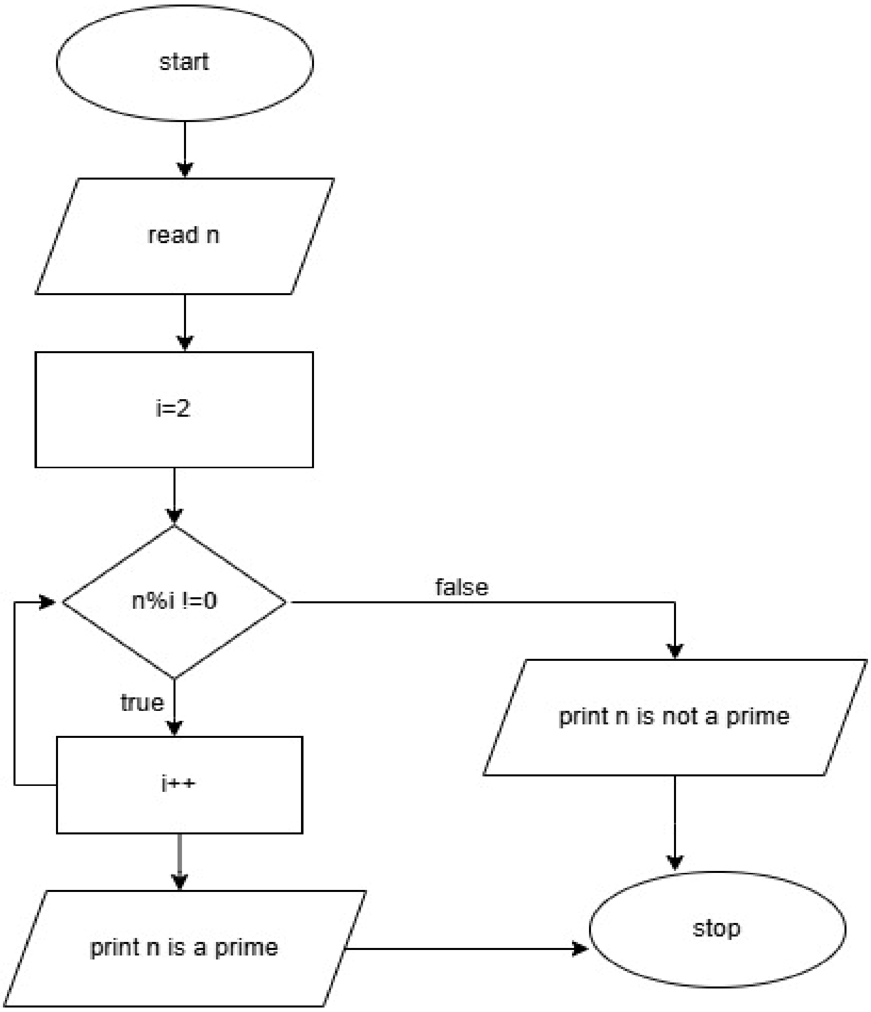
Set flag=0 Go to step 8

Step 7: i=i+1

Step 8: if flag==0, print “n is not prime number” Else print” n is prime number”

Step 9: Stop

Flowchart



Program

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

```
int i,n, m=0,flag=0;
 printf("Enter a number");
 scanf("%d",&n);
m=n/2;

 for(i=2;i<=m;i++)
{
if(n%i==0)
{
printf("Number is not prime");
 flag=1;
break;
}
}
if (flag==0) Printf("Numberr is prime");
 getch();
}
```

Output

```
Enter a number 29
Number is prime
```

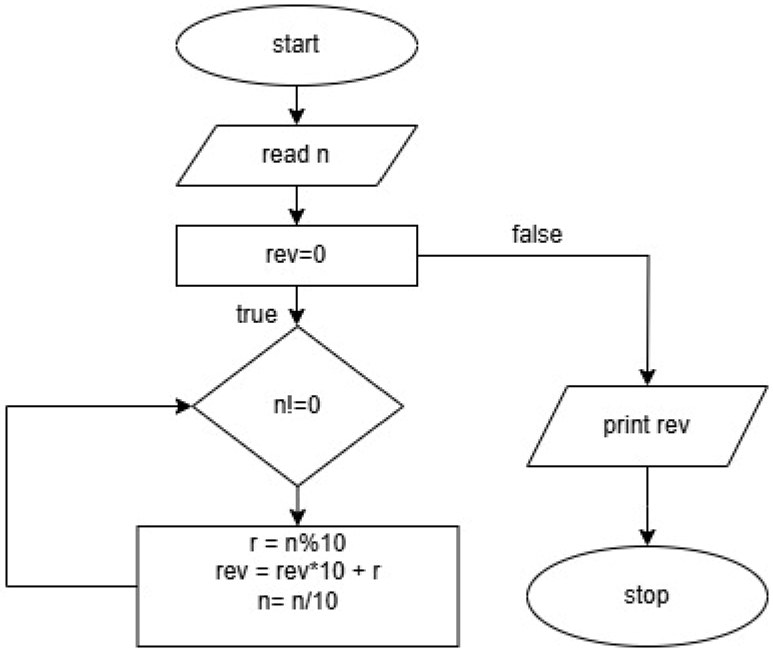
7. write a C program to find the reverse of a given number

Algorithm

```
Step 1: Start
Step 2: Declare variables n, rev, r
Step 3: Read n
Step 4: rev=0
Step 5: while(n!=0) r=n%10; rev=rev*10+r;
        n=n/10;
Step 6: Display rev
```

Step 7: Stop

Flowchart



Program

```
# include <stdio.h>
#include<conio.h>
void main()
{
int n, rev=0,r;
 printf("Enter an integer:");
scanf("%d",&n);
while (n!=0)
{
r=n%10;
rev =rev*10+r;
n=n/10;
}
printf("Reversed number =%d",rev);
getch();  }
```

Output

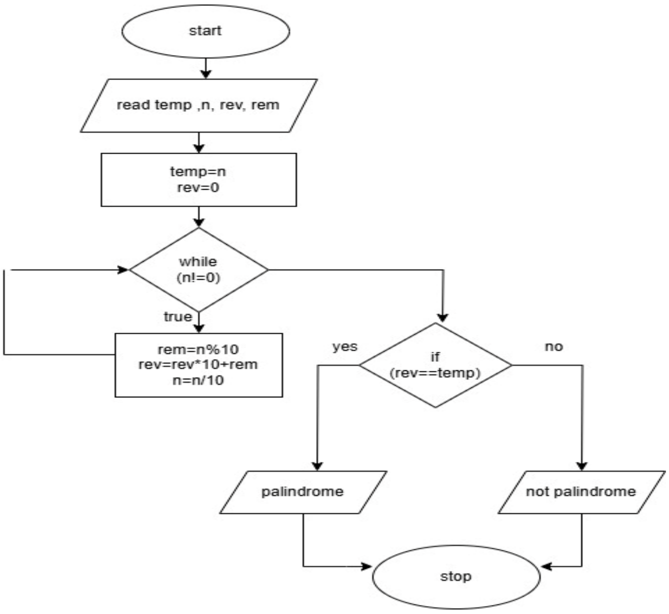
```
Enter an integer: 5432
 Reversed number= 2345
```

8. write a C program to check whether given number is palindrome or not

Algorithm

```
Step 1: Start
Step 2: Declare variables n,rev,temp,rem
Step 3: rev=0, temp=n
Step 4: while( n!=0) rem=n%10 rev =rev*10+rem n=n/10
Step 5: if (rev==temp) Print ‘it is a palindrome’
Else Print ‘it is not a palindrome’
 Step 6: Stop
```

Flowchart



Program

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

```
int n,rev=0,rem,temp;
 printf(“ Enter an Integer:”);
 scanf(“%d”,&n);
temp=n;
while(n!=0)
{
rem=n%10;
rev=rev*10+rem;
 n=n/10;
}
if(temp==rev)
printf(“%d is a palinmdrome :”,temp);
 else
printf(“%d is not a palindrome : “,temp);
 getch();}
```

Output

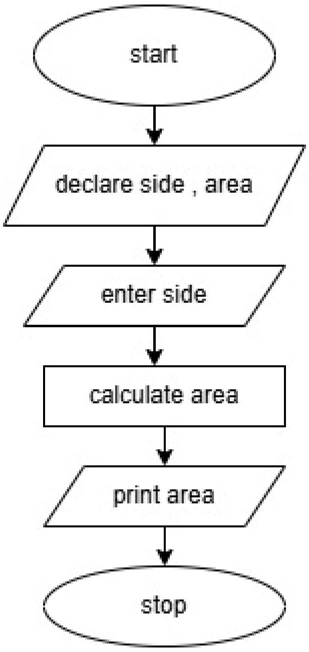
```
Enter an Integer: 1234321
1234321 is a palindrome
```

9. Write a C program to calculate area of equilateral triangle

Algorithm

```
Step 1: Start
Step 2: Declare variables side, area
Step 3: Read side
Step 4: Calculate area
Step 5: Print area
Step 6: Stop
```

Flowchart



Program

```
#include<stdio.h>
 #include<conio.h>
#include<math.h>
void main ()
{
float side,area;
 clrscr();
printf(“Enter the value of an equilateral triangle: ”);
 scanf(“%f”,&side);
area=sqrt((3)/4)*(side*side);
printf(“\n\n Area of an equilateral triangle:%f sq.unit”,area);
getch();
}
```

Output

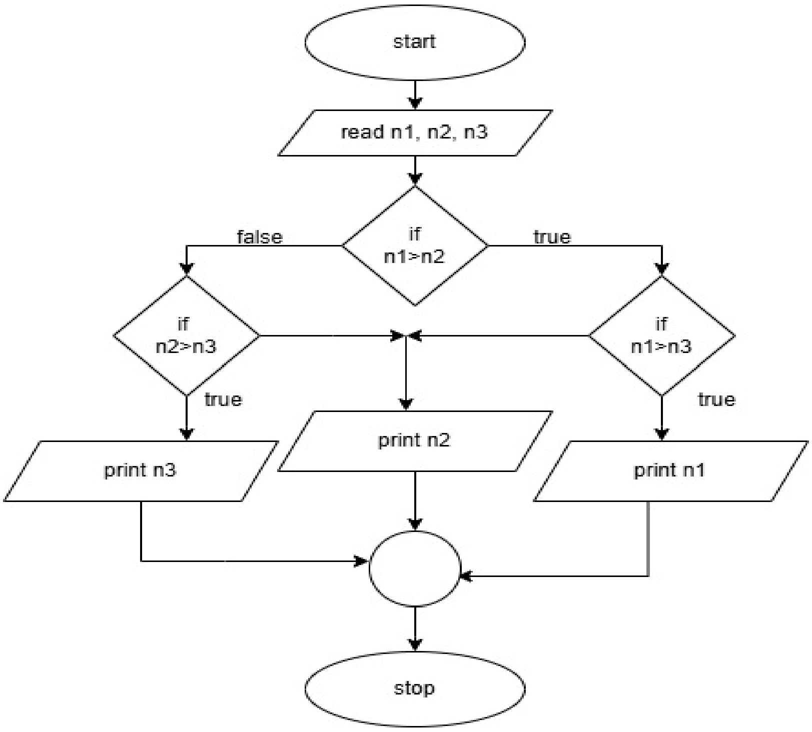
```
Enter the value of an equilateral triangle: 4
Area of an equilateral triangle:6.928203 sq.unit
```

10. write a C program to find the greatest of 3 numbers

Algorithm

```
Step 1: Start
Step 2: Declare variables n1, n2, n3
 Step 3: If n1>n2, then go to the step 6
Step 4: If n2>n3, then print n2 & go to step 8
Step 5: Print n3 is greatest & go to step 8
Step 6: If n1>n3, then print n1 is greatest &go to step 8
Step 7: Print n3 is greatest
Step 8: End
```

Flowchart



Program

```
# include <stdio.h>
 #include <conio.h>
 void main()
{
int n1,n2,n3;
printf("Enter three different numbers:");
scanf("%d %d %d",&n1,&n2,&n3);
 if(n1>=n2&&n1>=n3)
printf("%d is the largest number.",n1);
 if(n2>=n1&&n2>=n3)
printf("%d is the largest number.",n2);
 if(n3>=n1&&n3>=n2)
printf("%d is the largest number.",);
 getch();
}
```

Output

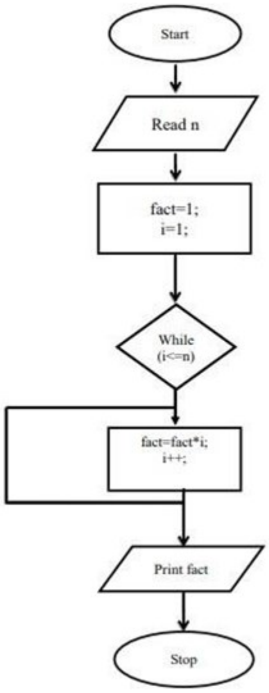
```
Enter three different numbers: 35 67 90
90 is the largest number
```

11. write a C program to find factorial of number without Recursion

Algorithm

```
Step 1: Start
Step 2: Declare variables n, i, fact
Step 3: Read n
Step 4: Initialize variables fact=1;
i=1;
Step 5: while(i<=n) fact=fact* i;
i=i++;
Step 6: Display fact
Step 7: Stop
```

Flowchart



Program

```
# include <stdio.h>
 #include <conio.h>
 void main()
{
int n,i,fact=1;
printf("Enter a number");
 scanf("%d",&n);
for(i=1;i<=n;i++)
{
fact=fact*i;
printf("factorial of %d is %d",n,fact);
 getch();
}
```

Output

```
Enter a number 5 Factorial of 5 is 120
```

12. Write a C program to find smallest element in an array

Algorithm

Step 1: Start

Step 2: Declare integer array a and define its values

Step 3: Set small to 0

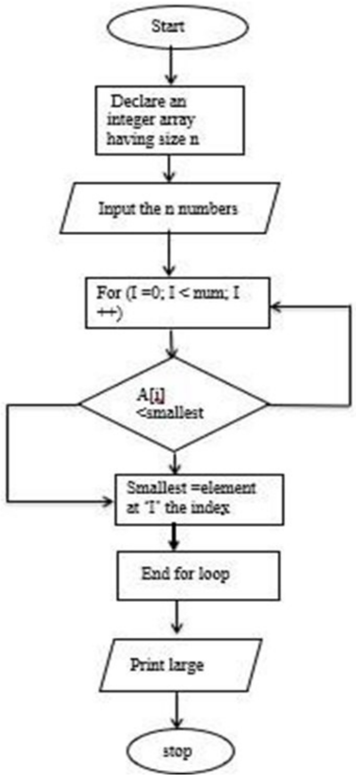
Step 4: Loop for each value a

Step 5: if a [i] <small. Assign a[i] to small

Step 6: After loop finishes, display large as largest element in an array

Step 7: Stop

Flowchart


Start
Declare an integer array having size n
Input the n numbers
For (I =0; I < num; I ++)
A[i] <smallest
Smallest =element at 'I' the index
End for loop
Print large
stop

Program

```
#include<stdio.h>

#include<conio.h>

void main()

{

int a[30], i, num, smallest;
```

```
 clrscr();
printf("\nEnter no of elements :");
scanf("%d", &num);
printf("\n element are:");
 for (i = 0; i<num; i++)
scanf("%d", &a[i]);
 smallest = a[0];
for (i = 0; i<num; i++)
 {
if (a[i] < smallest)
{
 smallest = a[i];
}  }
printf("\nSmallest Element : %d", smallest);
getch();
}
```

Output

```
Enter the number of Elements: 8    Elements are: 78
98
88
23
65
10
48
99
Smallest Element: 10
```

13. Write a C program to find largest element in an array

Algorithm

```
Step 1: Start
```

Step 2: Declare integer array ‘a’ and define its values

Step 3: Set large to 0

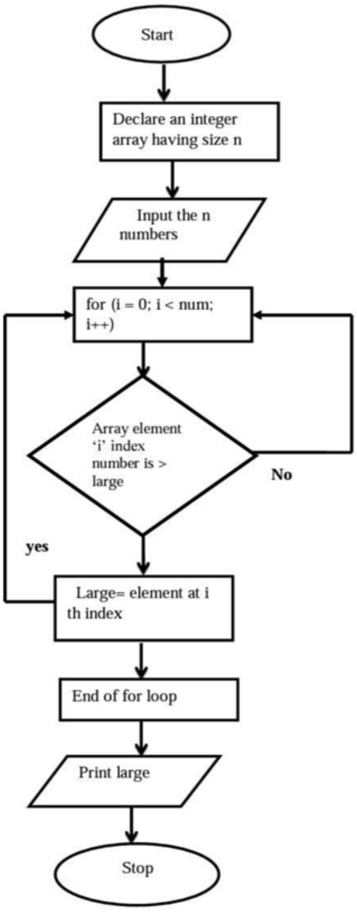
Step 4: Loop for each value a

Step 5: If a [i]>large, Assign a[i] to large

Stop 6: After loop finishes, Display large as largest element in an array

Stop 7: Stop

Flowchart


Start
Declare an integer array having size n
Input the n numbers
for (i = 0; i < num; i++)
Array element ‘i’ index number is > large
No
yes
Large= element at i th index
End of for loop
Print large
Stop

Program

```
 #include<stdio.h>
#include<conio.h>
 void main ()
{
int a [30], i, num, large;
 printf("\nEnter size of the array :");
scanf("%d", &num);
 printf("\nEnter the element :");
for (i = 0; i<num; i++)
scanf("%d", &a[i]);
large = a[0];
for (i = 0; i<num; i++)
{
if (a[i] > large)
{
large = a[i];
}
}
printf("\nLargest Element : %d", large);
 getch();
}
```

Output

```
Enter size of the array: 4    Enter the element: 678
 987
671
345
Largest Element: 987
```

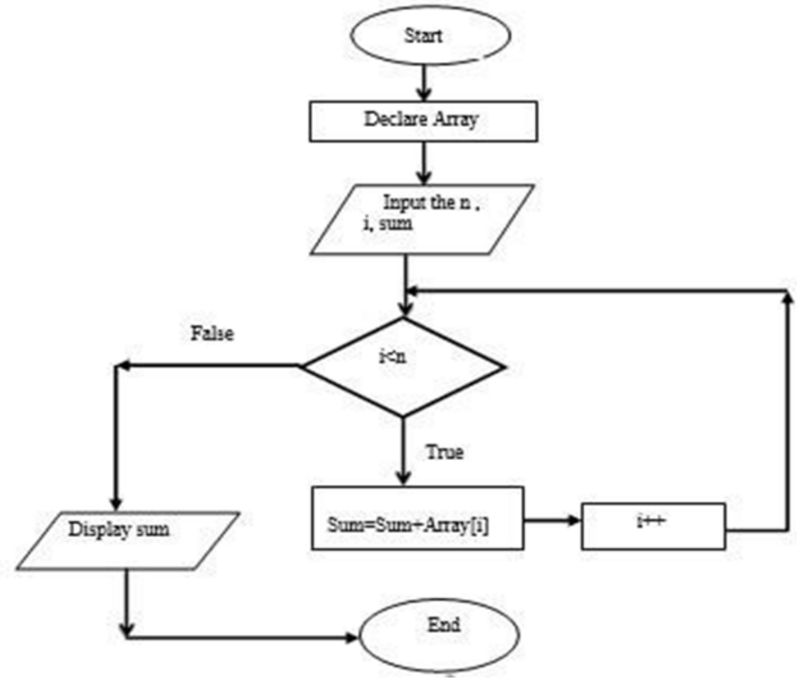
14. Write a C program to addition of all elements in an array.

Algorithm

```
Step 1: Start.
Step 2: Take an array A and define its values.
 Step 3: Loop for each value of A.
Step 4: Add each element to ‘sum’ variable.
 Step 5: After the loop finishes, display ‘sum’.
```

Step 6: Stop

Flowchart



Program

```
#include <stdio.h>
#include <conio.h>
void main()
{
int a[100];
int i, n, sum=0;
printf("Input the number of elements in the array :");
 scanf("%d",&n);
printf("Input %d elements in the array :\n",n);
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
for(i=0; i<n; i++)
{
sum += a[i];
}
printf("Sum of all elements stored in the array is : %d\n\n", sum);
getch();
}
```

Output

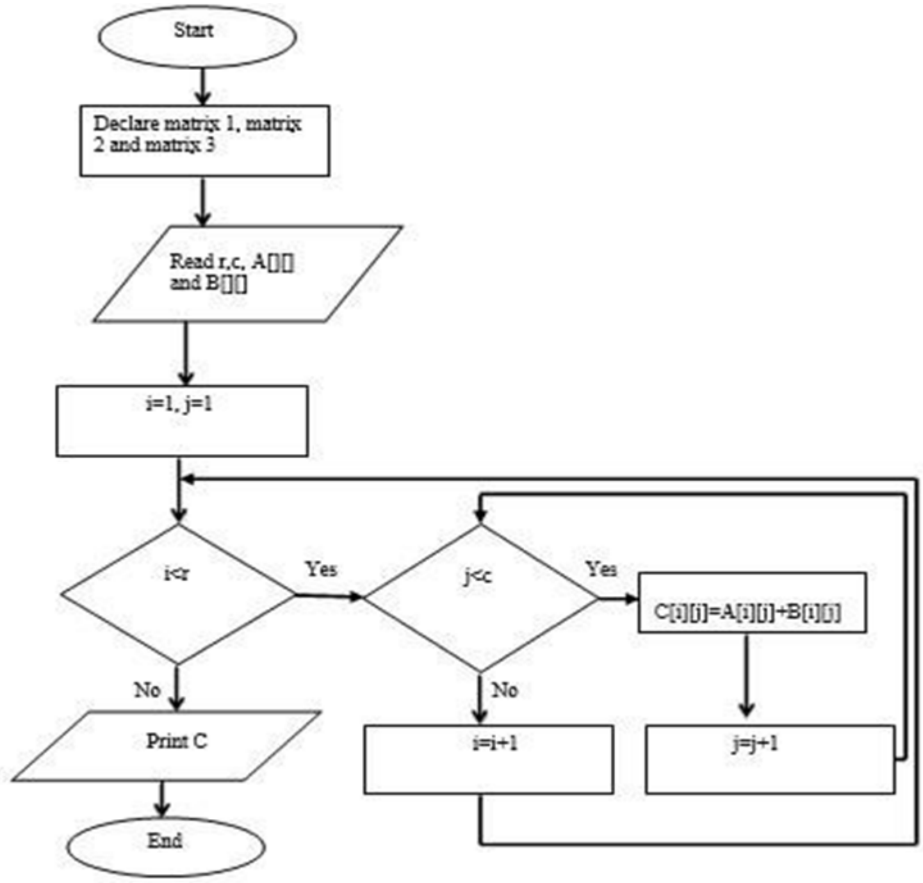
```
Input the number of elements in the array: 5   Input 5 elements in the array:
5
2
3
10
25
Sum of all elements stored in the array is: 45
```

15. Write a C program to addition of 2 matrices

Algorithm

```
Step 1: Start.
Step 2: Declare matrix 1[A[][]] and matrix 2[B[][]], r, c.
Step 3: Read r, c, A [][] and B [][].
Step 4: for i=1 to rows [matrix 1]. For j=1 to columns [matrix 1] Input matrix 1 [i,j] Input matrix 2 [i,j]
Matrix 3 [C[i],[j]]= matrix 1 [A[i],[j]] + matrix 2 [B[i],[j]].
Step 5: Display matrix 3 [C[r][c]].
Step 6: Stop.
```

Flowchart



Program

```
#include <stdio.h>
 #include <conio.h>
 void main()
{
int r, c, A[100][100], B[100][100], C[100][100], i, j;
printf("Enter the number of rows: ");
scanf("%d", &r);
printf("Enter the number of columns: ");
 scanf("%d", &c);
printf("\nEnter elements of 1st matrix:\n");
 for (i = 0; i< r; ++i)
for (j = 0; j < c; ++j)
{
d", &A[i][j]);
}
printf("Enter elements of 2nd matrix:\n");
 for (i = 0; i< r; ++i)
for (j = 0; j < c; ++j)
{
scanf("%d", &B[i][j]);
}
for (i = 0; i< r; ++i) for (j = 0; j < c; ++j)
{
C[i][j] = A[i][j] + B[i][j];
}
printf("\nSum of two matrices: \n");
 for (i = 0; i< r; ++i)
for (j = 0; j < c; ++j)
{
printf("%d ", C[i][j]);
 if (j == c - 1)
{
printf("\n\n");
} }
getch();
}
```

Output

```
Enter the number of rows: 3 Enter the number of columns: 2 Enter elements of 1st matrix:
3
4
5
6
7
8
Enter the element of 2nd matrix:
```

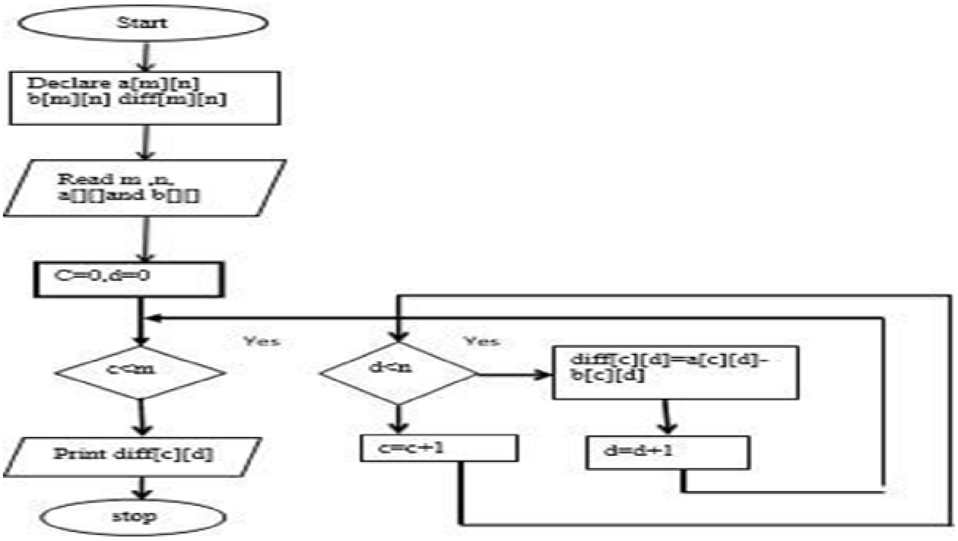
```
3
2
1
4
56
6
Sum of two matrices:
6 6
6 10
63 14
```

16. Write a C program to evaluate subtraction of two matrices

Algorithm

```
Step 1: Start
Step 2: Declare matrix A[m][n], matrix B[m][n] and matrix C[m][n]; m= no. of rows,
        n= no. of columns
Step 3: Read r, c, A[][] and B[][]
Step 4: Declare variable c=0, d=0
Step 5: Repeat until c < m
5.1  : Repeat until d< diff[c][d]=A[c][d] - B[c][d] Set d=d+1
5.2  : Set c=c+1
Step 6: diff is the required matrix after subtraction
Step 7: Stop
```

Flowchart


Start
Declare a[m][n] b[m][n] diff[m][n]
Read m ,n, a[][]and b[][]
C=0,d=0
Yes
c<m
Yes
d<n
diff[c][d]=a[c][d]-b[c][d]
Print diff[c][d]
c=c+1
d=d+1
stop

Program

```
#include <stdio.h>
#include<conio.h>
 void main()
{
int m, n, c, d, first[10][10], second[10][10], difference[10][10];
 clrscr();
printf("Enter the number of rows and columns of matrix\n");
scanf("%d%d", &m, &n);
printf("Enter the elements of first matrix\n");
for (c = 0; c < m; c++)
 for(d = 0 ; d < n; d++)
 scanf("%d", &first[c][d]);
printf("Enter the elements of second matrix\n");
 for (c = 0; c < m; c++)
for (d= 0; d <n;d++)
 scanf("%d",&second[c][d]);
printf("Difference of entered matrices:-\n");
for (c = 0; c < m; c++)
{
```

```
for (d = 0; d < n; d++)
{
difference[c][d] = first[c][d] - second[c][d];
printf("%d\t",difference[c][d]);
}
printf("\n");
}
getch();
}
}
```

Output

```
Enter the number of rows and columns of matrices 2
2
Enter the elements of first matrix 56
45
32
24
Enter the elements of second matrix 10
12
14
9
Difference of entered matrices: - 46 33
18 15
```

17. Write a C program to multiplication of 2 matrices

Algorithm

Step 1: Start.

Step 2: Enter the row and column of the first (A) matrix.

Step 3: Enter the row and column of the Second (B) matrix.

Step 4: Enter the elements of the first (A) matrix.

Step 5: Enter the elements of the second (B) matrix.

Step 6: Print the elements of the first (A) matrix in matrix form.

Step 7: Print the elements of the second (B) matrix in matrix form.

Step 8: Set a loop up to row.

Step 9: Set an inner loop up to the column.

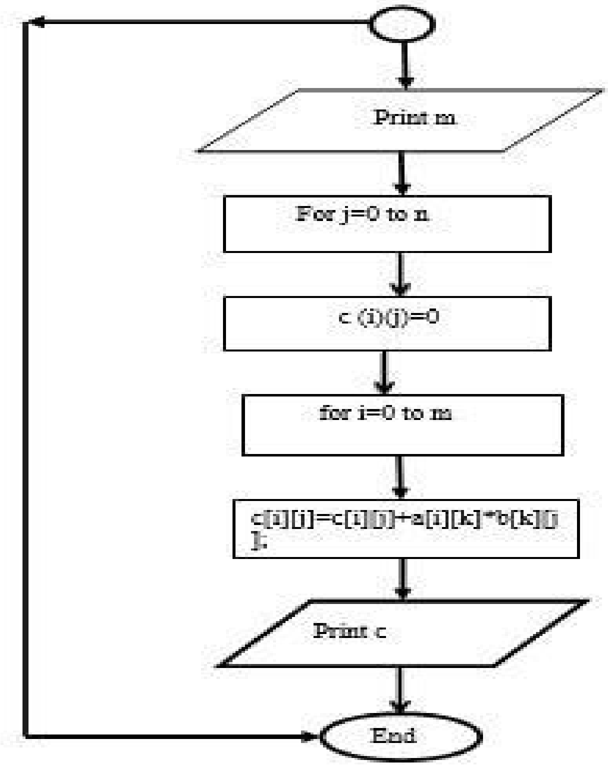
Step 10: Set another inner loop up to the column.

Step 11: Multiply the first (A) and second (B) matrix and store the element in the third matrix (C).

Step 12: Print the final matrix.

Step 13: Stop

Flowchart



Program

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

```
int a[25][25],b[25][25],c[25][25],i,j,k,r,s;
int m,n;
 clrscr();
printf("Enter the first matrix\n");
scanf("%d%d",&m,&n);
printf("Enter the second matrix\n");
 scanf("%d%d",&r,&s);
if(m!=r)
printf("\n The matrix cannot multiplied");
else
{
printf("\n Enter the elements of first matrix ");
for(i= 0;i<m;i++)
{
for(j=0;j<n;j++)
 scanf("\t%d",&a[i][j]);
}
printf("\n Enetr the elements of second matrix ");
 for(i=0;i<m;i++)
{
 for(j=0;j<n;j++)
 scanf("\t%d",&b[i][j]);
}
printf("\n The element of first matrix is");
 for(i=0;i<m;i++)
{
 printf("\n");
 for(j=0;j<n;j++)
 printf("\t%d",a[i][j]);
}
printf("\n The element of second matrix is");
```

```
 for(i=0;i<m;i++)
{
printf("\n");
 for(j=0;j<n;j++)
printf("\t%d",b[i][j]);
}
for(i=0;i<m;i++)
{
 printf("\n");
 for(j=0;j<n;j++)
{
c[i][j]=0;
for(k=0;k<m;k++) c[i][j]=c[i][j]+a[i][k]*b[k][j];
}
}
}
printf("\n Multiplication of two matrix is");
for(i=0;i<m;i++)
{
 printf("\n");
 for(j=0;j<n;j++)
 printf("\t%d",c[i][j]);
getch();
}  }
```

Output

```
Enter the first matrix 3
3
Enter the second matrix 3
3
Enter the elements of first matrix 45 32
12
```

```
67
87
65
66
78
90
Enter the element of second matrix 56 78
99
82
12
34
39
56
78
89
The element of first matrix is 45 32 12
67 87 65
66 78 90
The element of second matrix is 56 78 99
82 12 34
56 78 89
Multiplication of two matrix is 5816 4830 6611
14526 11340 15376
15132 13104 17196
```

18. Write a C program to find transpose of given square matrix

Algorithm

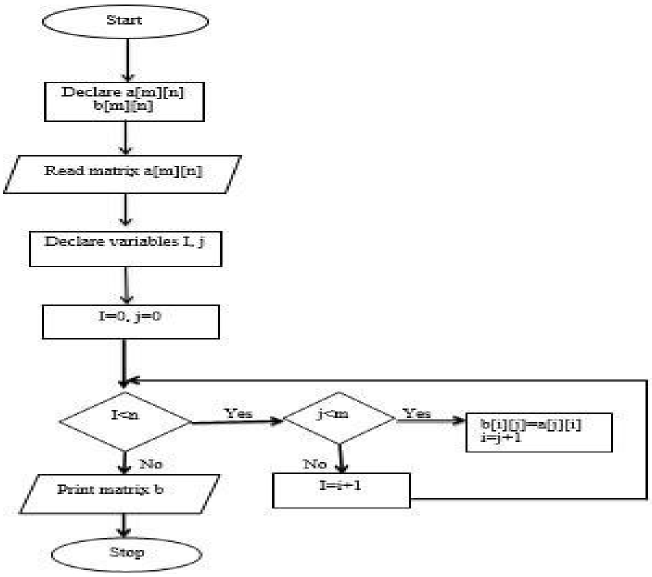
Step 1: Start

Step 2: Declare matrix a[m][n] of order m x n

Step 3: Read matrix a[m][n] from User

```
Step 4: Declare matrix b[m][n] of order m x n
 Step 5: Transposing the Matrix
5.1  : Declare variables i, j
5.2  : Set i=0, j=0
5.3  : Repeat until i< n
5.3.1: Repeat until j < m
5.3.1.1: b[i][j] = a[j][i]
5.3.1.2: j=j+1 // Increment j by 1
5.3.2: i=i+1 // Increment i by 1
5.4  : Print matrix b The matrix b is the transpose of a and can be printed now
 Step 6: Stop
```

Flowchart



Program

```
#include <stdio.h>
 #include<conio.h>
Void main()
{
int a[10][10], b[10][10], m, n, i, j;
Printf("Enter rows and columns: ");
 scanf("%d %d", &m, &n);
printf("\nEnter matrix elements:\n");
for ( i = 0; i< m; ++i)
for ( j = 0; j < n; ++j)
{
printf("Enter element a%d%d: ", i + 1, j + 1);
scanf("%d", &a[i][j]);
}
printf("\nEntered matrix: \n");
for ( i = 0; i< j; ++i)
for ( j = 0; j < n; ++j)
{
printf("%d ", a[i][j]);
 if (j == n - 1)
printf("\n");
}
for (i = 0; i< m; ++i)
for (j = 0; j < n; ++j)
{
b[j][i] = a[i][j];
}
printf("\nTranspose of the matrix:\n");
for ( i = 0; i< n; ++i)
for ( j = 0; j < m; ++j)
{
printf("%d ", b[i][j]); if (j == m - 1)
 printf("\n");
}
getch();
}
```

Output

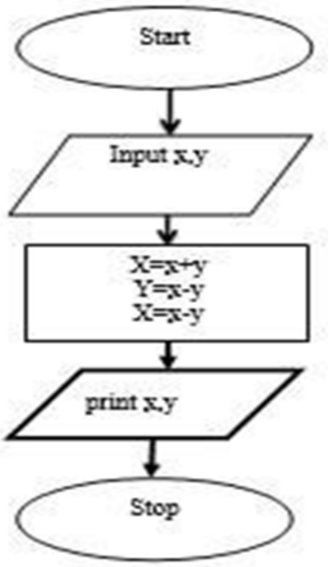
```
Enter the number of rows and columns: 2 2
Enter the matrix elements: Enter element a11:2
Enter element a12:4
Enter element a21:5
Enter element a22 : 6   Entered matrix:
2 4
5 6
Transpose of the matrix:
2 5
4 6
```

19. C program to Swap of 2 no’s without using third variable

Algorithm

```
Step 1: Start
 Step 2: Enter x,y
Step 3: Print x,y
Step 4: x=x+y
Step 5: y=x-y
Step 6: x=x-y
 Step 7: Print x,y
Step 8: Stop
```

Flowchart



Program

```
# include <stdio.h>
 # include<conio.h>
void main()
{
int x,y;
printf(“Enter first number:”);
scanf(“%d”,&x);
printf(“Enter second number:”);
 scanf(“%d”,&y);
x=x+y;
 y=x-y;
x=x-y;
printf (“\n after swapping ,first number =%d\n”,x);
Printf (“\n after swapping ,second number =%d\n”,y);
getch ();
}
```

Output

```
Enter first number: 10

Enter second number: 20

After swapping, first number = 20

After swapping, second number = 10
```

20. write a C program to implement stack operation using array

Algorithm for push operation

Step 1: Start

Step 2: Declare stack [MAX]

Step 3: Check if the stack is full or not by comparing top with (MAX-1) if the stack is full, then print “stack overflow”

Step 4: Else, the stack is not full, Increment top by 1 and set, a [top] =x Which pushes the element x into the address pointed by top

Step 5: Stop

Algorithm for pop () operation

Step 1: Start

Step 2: Declare stack [MAX]

Step 3: Print the element into the stack

Step 4: Check if the stack is empty or not by comparing with the base of

Array, i.e 0 if the top is less than 0, then stack is empty, Print” stack under flow”

Step 5: Else, If top is greater than zero the stack is not empty then the store the value pointed by top in a variable x=a [top] and decrement top by one .the popped element is x.

Step 6: Stop

Algorithm for Display () operation

Step 1: Start

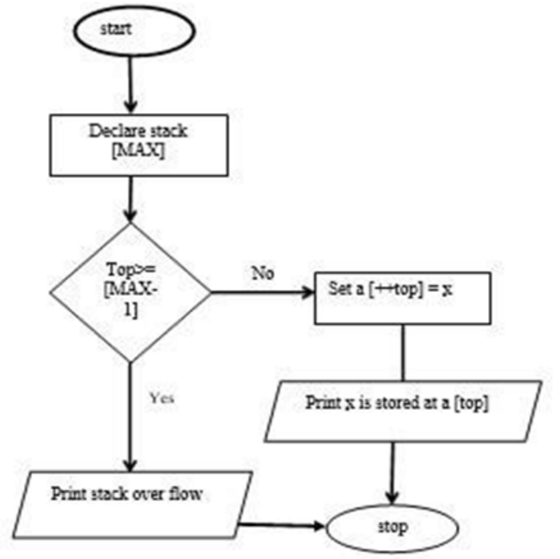
Step 2: Declare stack [Max]

Step 3: Push the element in to the stack

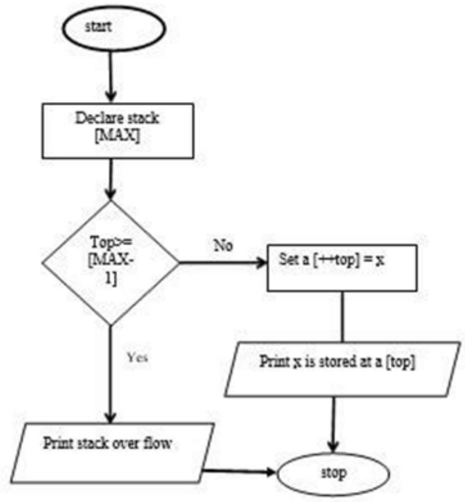
Step 4: Print the value stored into the stack pointed by top.

Step 5: Stop

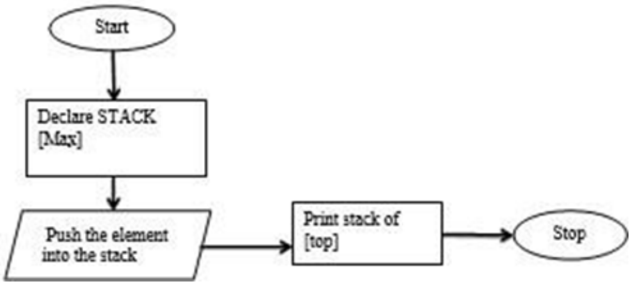
Flow chart for push()



Flow chart for pop() operation



Flow chart for display ()



program

```
#include <stdio.h>
 #include <conio.h>
 #define MAX 25 int a[MAX];
int top=-1;
void push(int elt)
{
if(top!=MAX-1)
{ top++;
 a[top]=elt;
printf("%d pushed\n",elt);
}
else
{
Printf ("stack full");
}
}
void pop()
{
if(top!=-1)
{
printf("\npoped%d",a[top]);
top--;
}
else
{
printf("stack empty");
}
}
void display()
{
int i; if(top!=-1)
{
printf("stack is");
```

```
 for(i=0;i<=top;i++)
printf("\t%d",a[i]);
}
}
void main()
{
int opt=0,n;
clrscr();
do
{
printf("1.push\n2.pop\n3.display\n4.exit\nenter your option");
 scanf("%d",&opt);
switch(opt)
{
case 1:printf("enter element top push\n");
 scanf("%d",&n);
push (n);
 break;
case 2 :pop();
 break;
case 3:display();
 break;
case 4:break;
 default:printf("invalid option");
}}
while(opt!=4);
 getch();
}
```

Output

1. Push

```
2. Pop
3. Display
4. Exit
Enter you option 1
Enter element top push 24 25 pushed
1. Push
2. Pop
3. Display
4. Exit
Enter your option 3 Stack is 24 25
```

21. Write a C program to print the pattern of binary numbers (pyramid)

Algorithm

Step 1: Start

Step 2: Take the number of rows(N) of right triangle as input from user using scanf function.
Step 3: Number of integers in Kth row is always K+1.

Step 4: We will use two for loops to print right triangle of binary numbers.

Step 5: Outer for loop will iterate N time (from 1 = 0 to N-1). Each iteration of outer loop will print one row of the pattern.

Step 6: Inside inner loop will toggle binary digits and print it. Each iteration of inner loop for Kth row will print K+1 alternating 0 and 1.

Step 7: Stop

Program

```
#include<stdio.h>
void main()
{
int i, j, rows;
int count = 1;
```

```
 clrscr();
printf("Enter the number of rows\n");
 scanf("%d", &rows);
for (i = 0; i< rows; i++)
{
for (j = 0; j <= i; j++)
{
printf("%d ", count);
count = !count;
}
count = i % 2;
 printf("\n");
}
getch();
}
```

Output

```
Enter the number of rows 5
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

22. write a C Program to perform stack operations using pointer

Algorithm

```
Step 1: Start
Step 2: Declare variables size,top,arr[MAX]
Step 3: Print 1.Push 2.Pop 3.Display 4. Exit
Step 4: void init_stk(struct stack *st)
Step 4: top=-1
Step 5: Stop void push()
Step 1: if (top<size) Top++; Print the pushed element
 Step 2: Else
Step 3: Print Stack is Overflow
 Step 3: END
Void pop()
Step 1: if(top==-1) Print stack is Underflow
 Step 2: Else
Step 3: num = st->arr[st->top] st->top—
 Step 4: END
Void display()
Step 1: Declare variable i
 Step 2: if (top==-1)
Print stack is underflow
Step 3: Else
Step 4: i=top
for (i = st->top; i>= 0; i--) print stack[i]
Step 5: END
```

Program

```
 #include<stdio.h>
 #include<conio.h>
 #include<stdlib.h>
 #define MAX 50 int size;
struct stack
{
int arr[MAX]; int top;
}
void init_stk(struct stack *st)
{
st->top = -1;
```

```
}
void push(struct stack *st, int num)
{
if (st->top == size - 1)
{
printf("\nStack overflow(i.e., stack full).");
 return;
}
st->top++;
st->arr[st->top] = num;
}
int pop(struct stack *st)
{
int num;
if (st->top == -1)
{
printf("\nStack underflow(i.e., stack empty).");
 return NULL;
}
num = st->arr[st->top];
st->top--;
return num;
}
void display(struct stack *st)
{
int i;
for (i = st->top; i>= 0; i--)
printf("\n%d", st->arr[i]);
}
void main()
{
```

```
int element, opt, val;
 struct stack ptr; init_stk(&ptr);
printf("\nEnter Stack Size :");
 scanf("%d", &size);
while (1)
{ printf("\n1.PUSH");
printf("\n2.POP"); printf("\n3.DISPLAY"); printf("\n4.QUIT");
printf("\n");
printf("\nEnter your option : ");
 scanf("%d", &opt);
switch (opt)
{
case 1:
printf("\nEnter the element into stack:");
 scanf("%d", &val);
push(&ptr, val); break;
case 2:
element = pop(&ptr);
printf("\nThe element popped from stack is : %d", element); break;
case 3:
printf("\nThe current stack elements are:"); display(&ptr);
break;
case  4:
exit(0); default:
printf("\nEnter correct option!Try again."); getch();
}
}  }
```

Output

```
Enter stack size: 4 1: PUSH
2: POP
3: DISPLAY
```

```
4: QUIT

Enter your option: 1

Enter the elements into stack: 34 1: PUSH

2: POP

3: DISPLAY

4: QUIT

Enter your option: 3

The current stack elements are: 34
```

23. write a C program to implement stack operations using singly linked list

Algorithm

```
Push()-Insert an element into the stack
 Step 1: Create a newNode with given value
Step 2: Check whether stack is Empty(top==NULL)
 Step 3: If it is Empty, then set newNode->next=NULL
Step 4: If it is not Empty,then set newNode->next=top
Step 5: Set top=newNode
POP()-Deleting an Element from the stack
Step 1: Check whether stack is EMPTY(top==NULL)
Step 2: if it is Empty, then display “Stack is Empty”and terminate the function
Step 3: if it is Not Empty, then define a node pointer ‘temp’ and set it to ‘top’
 Step 4: then set ‘top=top->next’.
Step 5: Delete ‘temp’ (free(temp)).
DISPLAY()-Displaying stack of elements
Step 1: Check whether stack is EMPTY(top==NULL)
Step 2: if it is Empty, then display “Stack is Empty”and terminate the function
Step 3: if it is Not Empty, then define a node pointer ‘temp’ and initialize it to ‘top’
Step 4: Display ‘temp->data->’and move it to the next node. Repeat the same until temp reaches to the first node in the stack (temp>next!=NULL)
```

Step 5: Display ‘temp->data->NULL’

Program

```
#include <stdio.h>
 #include <stdlib.h>
struct node
{
int info;
struct node *ptr;
}*top,*top1,*temp;
int topelement();
void push(int data);
 void pop();
void empty();
 void display();
 void destroy();
void stack_count();
 void create();
int count = 0; void main()
{
int no, ch, e;
printf("\n 1 - Push"); printf("\n 2 - Pop"); printf("\n 3 - Top"); printf("\n 4 - Empty");
printf("\n 5 - Exit"); printf("\n 6 - Dipslay"); printf("\n 7 - Stack Count"); printf("\n 8 -
Destroy stack");
create();
while (1)
{
printf("\n Enter choice : ");
scanf("%d", &ch);
switch (ch)
{
```

```
case 1:
printf("Enter data : ");
scanf("%d", &no);
push(no);
break;
 case 2:
pop();
break;
case 3:
if (top == NULL)
printf("No elements in stack");
else
{
e = topelement();
printf("\n Top element : %d", e);
}
break;
 case 4:
empty();
 break;
 case 5:
exit(0);
 case 6:
display();
 break;
case 7:
stack_count();
break;
case 8:
destroy();
break;
```

```
 default :
printf(" Wrong choice, Please enter correct choice "); break;
}
getch();
}
}
void create()
{
top = NULL;
}
void stack_count()
{
printf("\n No. of elements in stack : %d", count);
}
void push(int data)
{
if (top == NULL)
{
top =(struct node *)malloc(1*sizeof(struct node)); top->ptr = NULL;
top->info = data;
}
else
{
temp =(struct node *)malloc(1*sizeof(struct node)); temp->ptr = top;
temp->info = data;
= temp;
}
count++;
}
void display()
{
```

```
top1 = top;
if (top1 == NULL)
{
printf("Stack is empty"); return;
}
while (top1 != NULL)
{
printf("%d ", top1->info); top1 = top1->ptr;
}
}
void pop()
{
top1 = top;
if (top1 == NULL)
{printf("\n Error : Trying to pop from empty stack");
 return;
}
Else
top1 = top1->ptr;
printf("\n Popped value : %d", top->info);
 free(top);
top = top1;
count--;
}
int topelement()
{
return(top->info);
}
void empty()
{
if (top == NULL) printf("\n Stack is empty");
```

```
else
printf("\n Stack is not empty with %d elements", count);
}
void destroy()
{
top1 = top;
while (top1 != NULL)
{
top1 = top->ptr; free(top);
top = top1;
top1 = top1->ptr;
} free(top1);
 top = NULL;
printf("\n All stack elements destroyed");
 count = 0;
}
```

Output

```
 1-push
 2-pop
3-top
4-empty
 5-exit
6-display
7-stack count
8-Destroy stack Enter data: 345
Enter choice: 3
Top element: 345
Enter choice: 6
345
Enter choice: 7
No:of elements in stack: 1 Enter choice: 5ter choice: 1
```

24. write a C program to concat two strings with using library function: strcat

Algorithm

```
Step 1: Start
Step 2: Declare the variables str1[100], str2[100],str3[100],length
Step 3: Read the string1
Step 4: Read the string 2
Step 5: strcpy(str3,str1)
Step 6: strcat(str3,str2)
Step 7: Print the concatenated string
Step 8: Stop
```

Program

```
#include<stdio.h>
#include<string.h>
void main()
{
char str1[100];
char str2[100]; char str3[100]; int len;
printf("\nEnter the String 1 : ");
gets(str1);
printf("\nEnter the String 2 : ");
gets(str2);
strcpy(str3, str1);
strcat(str3, str2);
printf("\nConcated String : %s", str3);
getch();
}
```

Output

```
Enter the String 1: Hai

Enter the String 2:hello Concatedstring :Haihello
```

25. write a C program to find the length of string using library function

Algorithm

```
Step 1: Start
Step 2: Declare variable len
Step 3: Print the string to be entered
Step 4: print Length of Given String
Step 5: Stop
```

Program

```
#include<stdio.h>
 #include<string.h>
 void main()
{
char str[100];
int len;
printf("\nEnter the String : ");
gets(str);
len = strlen(str);
printf("\nLength of Given String : %d", len);
getch();
}
```

Output

```
Enter the string: Welcome Length of given string: 7
```

26. write a C Program to compute sum of array elements using pointers

Algorithm

```
Step 1: Start
Step 2: Declare variables i,n,sum=0,*pt
Step 3: Read the no:of array elements ntep 4: gets(str) len = strlen(str)
Step 4: Set pt=(int*) malloc(n*sizeof(int))
Step 5: Read the values of array
Step 6: Initialize i=0 to n
Step 7: Set sum=sum+*(pt+i)
Step 8: Print the value of sum
Step 9: Stop
```

Program

```
 #include<stdio.h>
 #include<conio.h>
void main()
{
int arr1[10];
int i,n, sum = 0; int *pt;
printf(" Input the size of the array");
scanf("%d",&n);
printf(" Input %d number of elements in the array : \n",n);
for(i=0;i<n;i++)
{
scanf("%d",&arr1[i]);
}
pt = arr1;
for (i = 0; i< n; i++)
{
sum = sum + *pt; pt++;
}
printf(" The sum of array is : %d\n\n", sum);
getch();
}
```

Output

```
Input size of the array 3
Input 3 number of elements in the array: 4
6
8
The sum of array is: 18
```

27. write a C Program to find length of the string using pointer

Algorithm

```
Step 1: Start
Step 2: Declare the variables i,*pt
Step 3: Read the String
Step 4: Call the function Length=string_in(str);
Step 5: Print the value length
Step 6: Stop
IStep 1: Declare count=0;
Step 2: while (*p!='\0') Count++
P++
Step 3: return count
Step 4:
Stop
```

Program

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

```
char str[20], *pt;
int i = 0;
printf("Enter Any string: ");
gets(str);
pt = str;
while (*pt != '\0') i++;
pt++;
}
printf("Length of String : %d", i);
getch();
}
```

Output

```
Enter Any string: Haii welcome Length of String: 12
```

28. write a C Program to concat two strings without using library function

Algorithm

```
Step 1: Start
Step 2: Declare variables str1[25],str2[25],i=0,j=0
Step 3: Read the first string
Step 4: Read the second string
Step 5: Call the function Concat(str1,str2)
Step 6: Print the Concatenated String
Step 7: Stop
int string_In(char str[])
Step 1: Declare i=0, j=0
Step 2: while (str1[i]!='\0') i++
Step 3: END
Void concat(char str[],char str2[])
Step 1: Declare variables i,j
Step 2: while (str2[j]!='\0')
Step 3: str1[i]=str2[j]; j++i++
Step 4: Set str1[i]='\0'
Step 5: END
```

Program

```
#include<stdio.h>
#include<conio.h>
void main(void)
{
char str1[25],str2[25];
int i=0,j=0;
printf("\nEnter First String:");
gets(str1);
printf("\nEnter Second String:");
gets(str2);
while(str1[i]!='\0')
i++;
while(str2[j]!='\0')
{
str1[i]=str2[j]; j++;
i++;
}
str1[i]='\0';
printf("\nConcatenated String is %s",str1);
}
}
```

Output

```
Enter first string: Hello Enter second string: hai Concatenated String is Hellohai
```

29. write a C Program to compare two strings without using Library Function [strcmp]

Algorithm

```
Step 1: Start
Step 2: Declare variables str1[30],str2[30],i
Step 3: Read 2 strings
Step 4: Set i=o while (str1[i] == str2[i] && str1[i] != '\0') i++;
if (str1[i] > str2[i])
Step 5: Print str1 > str2 else if (str1[i] < str2[i])
Print str1 < str2
 Step 6: Else Print str1 = str2
Step 7: Stop
```

Program

```c
#include<stdio.h>
#include<conio.h>
void main()
{
char str1[30], str2[30]; int i;
printf("\nEnter two strings :");
gets(str1);
gets(str2); i = 0;
while (str1[i] == str2[i] && str1[i] != '\0')
i++;
if (str1[i] > str2[i])
printf("str1 > str2");
else if (str1[i] < str2[i])
printf("str1 < str2");
else
printf("str1 = str2");
```

```
 getch();
}
```

Output

```
Enter two strings: haii hello Welcome
Str1<str2
```

30. write a C Program to calculate area of circle using pointer

Algorithm

```
Step 1: Start
Step 2: void area (float, float*)
Step 3: Declare variables radius, area1
Step 4: Print radius of circle
Step 5: area (radius, &area1)
Step 6: Print Area of Circle *a = 3.14 * r * r
Step 7: Stop
```

Program

```
#include<stdio.h>
#include<conio.h>
void area(float, float*);
void main()
{
float radius,area1;
printf("Enter radius of Circle\n");
scanf("%f", &radius);
area(radius, &area1);
printf("\nArea of Circle = %0.2f\n", area1);
getch();
}
```

```
void area(float r, float *a)
{
*a = 3.14 * r * r;
}
```

Output

```
Enter radius of circle 4 Area of circle =50.24
```

31. write a C program to compute sum of the array elements using pointers

Algorithm

```
Step 2: Declare variables I, n, sum=0, *a
Step 3: Read the
Step 1: Start
number of array elements n
Step 4: Set a=(int *) malloc(n*sizeof(int))
Step 5: Read the values of array
Step6: Initialize i=0 to n
Step 7: Set sum=sum+*(a+i)
Step 8: Print the value of sum
Step 9: Stop
```

Program

```
#include<stdio.h>
#include <conio.h>
void main()
{
int i,n,sum=0; int *a;
clrscr();
printf("\n\nEnter the size of the array");
scanf("%d",&n);
a=(int *) malloc(n*sizeof(int));
printf("\n\nEnter the elements of the array");
for (i=0;i<=n;i++)
{
scanf("%d",a+i);
}
for(i=0;i<=n;i++)
{
sum=sum+*(a+i);
}
printf("\n\nSUM OF ALL ELEMENTS IN ARRAY=%d\n",sum);
getch();
}
```

Output

```
Enter the size of the array 5
Enter the elements of the array 10 20 50 15 5
SUM OF ALL ELEMENTS IN ARRAY= 100
```

32. write a C program to read integers into an array and reversing them using pointers

Algorithm

```
Step 1: Start
Step 2: Declare variables size,I,arr[50], *ptr
Step 3: Set ptr=&arr[0]
Step 4: Read the number of array elements size
Step 5: Read the values of array
Step 6: Set ptr=&arr[size-1]
Step 7: Initialize i=size to 0
Step 8: Print the value of *ptr
Step 9: Stop
```

program

```
#include <stdio.h>
#include<conio.h>
#define max 50
void main()
{
int size, i, arr[max];
int *ptr;
clrscr();
ptr=&arr[0];
printf("\n\n Enter the size of the array");
scanf("%d",&size);
printf("\n\n Enter the elements of the array");
for(i=0;i<=size;i++)
{
scanf("%d",ptr); ptr++;
}
ptr=&arr[size-1];
printf("\n\n ARRAY IN REVERSE ORDER:");
for(i=size-1;i>=0;i--)
{
printf("%d\t",*ptr);
ptr--;
}
getch();
}
```

Output

```
Enter the size of the array 5

Enter the elements of the array 12 13 14 15 16
```

```
ARRAY IN REVERSE ORDER: 16 15 14 13 12
```

33. write a C Program to find even number pyramid

Algorithm

```
Step 1: Start
Step 2: Declare variables i, j, k
Step 3: for (i=1; i<=5; i++)
Step 4: for (j=1,k=2; j<=i; j++,k=k+2)
Step 5: Print k,n
Step 6: Stop
```

Program

```c
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,k;
for(i=1; i<=5; i++)
{
for(j=1,k=2; j<=i; j++,k=k+2)
{
printf(" %d",k);
}
printf("\n");
}
getch();
}
```

Output

```
2
```

```
2 4
2 4 6
2 4 6 8
2 4 6 8 10
```

34. write a C Program to print the double pyramid pattern how to write C program to print FLOYD’S triangle in C programming

Algorithm

```
Step 1: Start
Step 2: Declare variables out,in,row size
Step 3: Read the row size
Step 4: for (out=row_size;out>=-row_size;out)
Step 5: for (in=0;in<=abs(out);in++)
Step 6: Print *
Step 7: Stop
```

Program

```
#include<stdio.h>
#include<math.h>
void main()
{
int out, in;
int row_size;
printf("Enter the row size:");
scanf("%d",&row_size);
for(out=row_size;out>=-row_size;out--)
{
for(in=0;in<=abs(out);in++)
{
```

```
printf("*");
}
printf("\n");
}
getch();
}
```

Output

```
Enter the row size: 3
****
***
**
*
**
***
****
```

35. write a C program to copy text form one file to other file

Algorithm

Step 1: Start

Step 2: Declare variables and files pointers

Step 3: Assign file pointers to fopen() function with read format

Step 4: Assign file pointers to fopen() function with write format

Step 5: if file is not open print error message

Step 6: else file content is copied from one file to another using loop

Step 7: Close the file

Step 8: Stop

Program

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

```
#include<conio.h>
#include<process.h>
void main()
{
FILE *fp; FILE *fpt; char ch;
clrscr();
fp=fopen("data.txt","r");
fpt=fopen("copy.txt","w");
if(fp==NULL)
{
printf("cannot open file");
exit(1);
}
else if(fpt==NULL)
{
printf("can not open file");
exit(1);
}
else
{
do
{
ch=fgetc(fp);
fputc(ch,fpt);
}while(ch!=EOF);
printf("\n\n FILE COPIED");
}
fcloseall();
printf("\n\n contents read from first file:\n\n");
fp=fopen("data.txt","r");
while(!feof(fp))
printf("%c",getc(fp));
printf("\n\n contents read from second file:\n\n");
fpt=fopen("copy.txt","r");
while(!feof(fpt))
printf("%c",getc(fpt));
fcloseall();
getch();
}
```

Output

```
FILE COPIED
Content read from first file:
Hello
Content read from second fileHello
```

36. Write a program to check whether the number is even or odd

Algorithm

Step 1: START.

Step 2: Enter the Number to Check.

Step 3: if Number is divisible by 2 then the Number is Even.

Step 4: Else Number is Odd.

Step 5: Print the Output.

Step 6: STOP

Program

```
#include <stdio.h>
int main()
{
int n;
printf("Enter a Number to Check Even or Odd\n");
scanf("%d", &n);
if (n%2 == 0)
printf("The Number is Even\n");
else
printf("The Number is Odd\n");
return 0;
}
```

Output

```
Enter a Number to Check Even or Odd:  7
The Number is Odd
```

37. Write a program to check whether the number is positive or negative.

Algorithm

```
Step1: start
Step 2: Num<-- 0.
Step 3: Read Num.
Step 4: Is (Num> 0) Then. Begin.
Print “Positive” End.
Else if (Num<0) Then. Begin. Print “Negative” End. Else. Begin. Print “Zero” End.
Step 5: stop
```

Program

```
 #include<stdio.h>
int main()
{
int num; scanf(“%d”,&num);
if(num> 0)
printf(“Positive”);
else
{
```

```
printf("Negative"); return 0;
}
```

Output

```
44
Positive
```

38. Write a program to add two numbers

Algorithm

```
Step 1: start the program
Step2: Read the values of 'a&b'
Step 3: Compute the sum of the entered numbers 'a','b,c=a+b.
Step 4: Print the value of 'c'.
Step 5:Stop the program.
```

Program

```
#include<stdio.h>
int main()
{
int a, b ,sum;
printf ("\n Enter two no:");
scanf("%d%d",&a,&b);
sum=a+b; printf("Sum:%d", sum);
return(0);
}
```

output

```
Enter two number:5 6 Sum:11
```

39. Write a program to find largest of two numbers

Algorithm

```
Step 1: Start
Step 2: Read a, b
Step 3: If a>b then
Display “a is the largest number”. Otherwise
Display “b is the largest number”.
Step 4: Stop.
```

Program

```
#include<stdio.h>
int main()
{
int num1, num2;
printf(“Please Enter Two different values \n”);
scanf(“%d%d”,&num1,&num2);
if(num1> num2)
{
Printf(“%d is largest \n”,num1);
}
else if(num2> num1)
{
Printf(“%d is largest \n”,num2);
}
else
{
printf(“Both are equal \n”);
}
return 0;
}
```

Output

```
Please Enter Two different values:27 6 27 is Largest
```

40. Write a program to find simple interest

Algorithm

```
Step 1: start.
Step 2: Declaration of variable with their data type, like :- int P, R, T; float I
Step 3: Input the value in variable
Step 4: Arithmetic operator used to perform SI=(p*r*t)/100;
Step 5: printf( ) called to print value of variable
Step 6: stop
```

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
int p,r,t; float i;
printf("Enter the Principal, Rate and Time\n");
scanf("%d %d %d",&p,&r,&t);
i=p*r*t/100;
printf("simple interest is : %f",i);
getch();
}
```

Output

```
Enter the Principal, Rate and Time 5500
5
8
simple interest is : 2200.000000
```

41. Write a c program to check whether the number Armstrong or not

Algorithm

```
Step 1: Start
Step 2: Declare Variable sum, temp, num
Step 3: Read num from User
Step 4: Initialize Variable sum=0 and temp=num
Step 5: Repeat Until num>=0
5.1  sum=sum + cube of last digit i.e [(num%10)*(num%10)*(num%10)]
5.2  num=num/10 Step 6: IF sum==temp
Print "Armstrong Number" ELSE
Print "Not Armstrong Number"
Step 7: Stop
```

Program

```
#include <stdio.h>
int main()
{
int num, originalNum, remainder, result = 0;
printf("Enter a three-digit integer: ");
scanf("%d", &num);
originalNum = num;
while (originalNum != 0)
{
remainder = originalNum % 10;
result += remainder * remainder * remainder;
originalNum /= 10;
}
if (result == num)
printf("%d is an Armstrong number.", num);
else
printf("%d is not an Armstrong number.", num);
return 0;
}
```

Output

```
Enter a three-digit integer: 555
555 is not an Armstrong number
```

42. Write a program to check whether the number is perfect or not.

Algorithm

```
step1. Start
step2. Read n
step3. Initialize s=0
step4. for i=1 to n do
a.if(n%i)==0, then
b.s=s+i
step5. if s==n
then Print "Given Number is Perfect Number". Goto Step 7
step6. Print "Given Number is Not a Perfect Number"
step7. Stop
```

program

```
#include<stdio.h>
int main(){
int n,i=1,sum=0;
printf("Enter a number: ");
scanf("%d",&n);
while(i<n)
{
if(n%i==0)
sum=sum+i;
i++;
}
if(sum==n)
printf("%d is a perfect number",i);
else
printf("%d is not a perfect number",i); return 0;
}
```

Output

```
Enter a number: 6
6 is a perfect number
```

43. Write a program to find the roots of a quadratic equation

Algorithm

```
Step-1 :Start
Step-2 :Input A,B,C
Step-3 :DISC= B2 – 4 A * C
Step-4 :IF (DISC < 0) THEN Write Roots are Imaginary Stop ENDIF
Step-5 :IF (DISC==0) THEN Write Roots are Real and Equal
X1 = - B/(2*A) Write Roots are X1,X1 Stop ENDIF
Step-6 :IF (DISC >0) Write Roots are Real and Unequal X1= (- B + SQRT(DISC)) / (2*A)
X2= (- B + SQRT(DISC)) / (2*A)
Write Roots are X1,X2 Stop ENDIF
```

Program

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

```
Int a,b,c;
float root1,root2,disc;
clrscr();
printf("enter the value of a,b, andc");
scanf("%d%d%d",&a,&b,&c);
disc=(b*b)- (4*a*c);
if(disc<0)
{
printf("The discriminant is imaginary and roots are complex");
}
else if (disc==0)
{
root1=b/(2*a)
printf("The roots are equal, it is %2f',root1);
}
else if(disc>0)
{
root1=(-b+sqrt(disc)/ (2*a); root2 =(-b- sqrt(disc) /(2*a);
printf("The roots are %2f and %2f',root1,root2);
}
getch();
}
```

Output

```
enter the value of a, b and c: 2.3 4
5.6
root1 = -0.87+1.30i and root2 = -0.87-1.30i
```

44. Write a c program for calculator using switch statement

Algorithm

1 Step: BEGIN.

2 Step: PRINT ENTER YOUR CHOICE.

3 Step: ENTER YOUR CHOICE.

4 Step: ENTER TWO OPERANDS FOR OPERATION.

5 Step: USER WILL ENTER +,-,*,/ .

6 Step: SWITCH(OPERATOR)

7 Step: DO THE OPERATION.

8 Step: PRINT THE RESULT.

9 Step: EXIT.

Program

```
#include<stdio.h>
int main()
{
int choice;
long num1, num2, x;
printf("Please choose your option:"
"\n1 = Addition" "\n2 = Subtraction"
"\n3 = Multiplication" "\n4 = Division"
"\n5 = Squares" "\n6 = exit" "\n\nChoice: ");
scanf("%d", &choice);
//while loop check whether the choice is in the given range while(choice < 1 || choice > 6)
{
printf("\nPlease choose the above mentioned option." "\nChoice: ");
scanf("%d", &choice);
}
switch (choice)
{
case 1:
printf("Enter two numbers: \n");
scanf("%ld %ld", &num1, &num2);
x = num1 + num2;
```

```
printf("Sum = %ld", x);
break;
case 2:
printf("Enter two numbers: \n");
scanf("%ld %ld", &num1, &num2); x = num1 - num2;
printf("Subtraction = %ld", x);
break;
case 3:
printf("Enter two numbers: \n");
scanf("%ld %ld", &num1, &num2);
x = num1 * num2;
printf("Product = %ld", x);
break;
case 4:
printf("Enter Dividend: ");
scanf("%d", &num1);
printf("Enter Divisor: ");
scanf("%d", &num2);
//while loop checks for divisor whether it is zero or not while(num2 == 0)
{
printf("\nDivisor cannot be zero." "\nEnter divisor once again: ");
scanf("%d", &num2);
}
x = num1 / num2; printf("\nQuotient = %ld", x);
break;
case 5:
printf("Enter any number: \n");
scanf("%ld", &num1);
x = num1 * num1;
printf("Square = %ld", x);
break;
```

```
case 6:
return;
default: printf("\nError");
}
}
```

Output

```
Pls choose your option 1=addition 2=substraction 3=multiplication 4=division
5=squares 6=exit Choice:1
Enter two numbers:
5
5
Sum=10
```

45. write a C program for find whether a year is a Leap Year or not.

Algorithm

```
Step1: Start the program
Step2: Declare year as int data type
Step3: Read the year
Step4: Check if (ye%4==0) Print "It is a leap year"
Step5: Else
Step5.1: Print "It is not a leap year"
Step6: stop the program.
```

Program

```
#include<stdio.h>
void main()
{
int year;
printf("Enter Year to check : ");
scanf("%d",&year);
if(((year%4==0)&&(year%100!=0))||(year%400==0))
printf("%d is a Leap Year",year);
else
printf("%d is not a Leap Year",year);
}
```

Output

```
Enter year to check:2004 2004 is a leap year
```

46. write a C program to print the Fibonacci series.

Algorithm

1. Read the value r.
2. Assign a=0, b=1 and c=0.
3. Print the value of a and b.
4. Compute c= a+b.
5. Check the condition (c<=r),goto step 6.Otherwise step 7.
6. Print the value of c.
7. Assign a=b and b=c.
8. Repeat the steps 4 to 7until (c<=r)

Program

```
#include <stdio.h>
int main()
{
int Number, i = 0, Next, First_Value = 0, Second_Value = 1;
printf("\n Please Enter the Range Number: ");
scanf("%d",&Number);
while(i< Number)
{
 if(i<= 1)
{
 Next = i;
}
else
{
Next = First_Value + Second_Value;
First_Value = Second_Value;
Second_Value = Next;
}
printf("%d \t", Next);
i++;
}
return 0;
}
```

Output

```
Please enter the range number:5
0    1 1 2 3
```

47. Write a program to find sum of digits

Algorithm

```
Step 1: Input N
Step 2: Sum = 0
Step 3: While (N != 0)
Rem = N % 10;
Sum = Sum + Rem; N = N / 10;
Step 4: Print Sum
```

Program

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

```
int main()
{
int n, remainder, sum = 0;
printf("Enter a number: ");
scanf("%d", &n);
while(n != 0)
{
remainder = n % 10;
sum += remainder;
n = n / 10;
}
printf("sum = %d", sum);
return 0;
}
```

Output

```
Enter a number: 12345 sum = 15
```

48. Write an interactive program toconvert decimal number to its hexadecimal equivalent.

Algorithm

1. Start
2. Initialize decimalnum,quotient,remainder,i and j
3. Enter a decimal number
4. Check WHILE condition
5. Print the result
6. Stop

Program

```
#include&lt;stdio.h&gt;
#include&lt;conio.h&gt;
```

```
void main()
{
long decimalnum,quotient,remainder;
int i,j=0;
clrscr();
char hexadecimalnum [100];
printf(&quot;\n\tEnter decimal number : &quot;);
scanf (&quot;%ld&quot;, &amp;decimalnum);
quotient=decimalnum;
while(quotient!=0)
{
remainder=quotient%16;
if(remainder&lt;10)
hexadecimalnum [j++] = 48+remainder;
else
hexadecimalnum[j++]=55+remainder;
quotient = quotient/16;
}
for (i=j;i&gt;=0;i--)
printf(&quot;\n\t\t\t%c&quot;,hexadecimalnum[i]);
getch();
}
```

Output

```
Enter decimal number: 255
Hexadecimal equivalent: FF
```

49. Write an interactive C program to find the MINIMUM and MAXIMUM (value) array elements in a given 3X3 matrix.

Algorithm

Step 1 : Start

Step 2 : Declare an array and all other necessary variables.

Step 3 : Read the size and store that value into the variable

Step 4 : Read the elements using scanf and store the entered array elements into the array using for loop

Step 5 : Initialize min,max values with the 1st element of the array

Step6 : Compare min,max values with a[i]

Step 7 : Print the minimum and maximum of the array values

Step 8 : Stop

Program

```
#include<stdio.h>
#include<conio.h>
int main()
{
int a[1000],i,n,min,max;
printf("Enter size of the array:");
scanf("%d",&n);
printf("Enter elements in array:");
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
min=max=a[0];
for(i=1;i<n;i++)
{
if(min>a[i])
min=a[i];
if(max<a[i])
max=a[i];
}
printf("Minimum of array is: %d",min);
printf("\nMaximum of array is : %d",max);
getch();
return 0;
}
```

Output

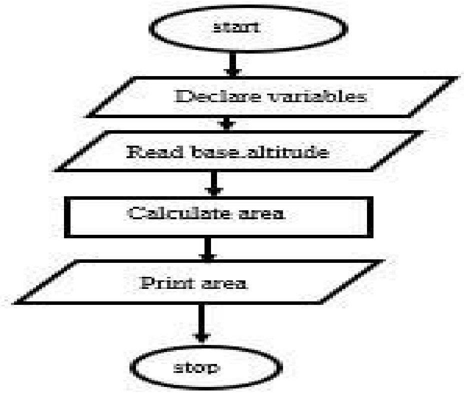
```
Enter size of the array: 5
Enter elements in array: 12 45 7 89 23
Minimum of array is: 7
Maximum of array is: 89
```

50. Write a C program to calculate area of right angled triangle

Algorithm

```
Step 1: Start
Step 2: Declare variables base, altitude, area
Step 3: Read base, altitude
Step 4: Calculate area
Step 5: Print area
Step 6: Stop
```

Flowchart



Program

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

```
#include<conio.h>
void main()
{
int base, altitude;
 float area;
clrscr();
printf(“Enter base of triangle: ”);
scanf(“%d”,&base);
printf(“Enter altitude of triangle: ”);
 scanf(“%d”, &altitude);
area=(base*altitude)/2;
printf(“\n\n Area of right angled triangle:%f ”,area);
getch();
}
```

Output

```
Enter base of triangle: 2
 Enter altitude of triangle: 2
Area of right-angled triangle: 2.0000
```

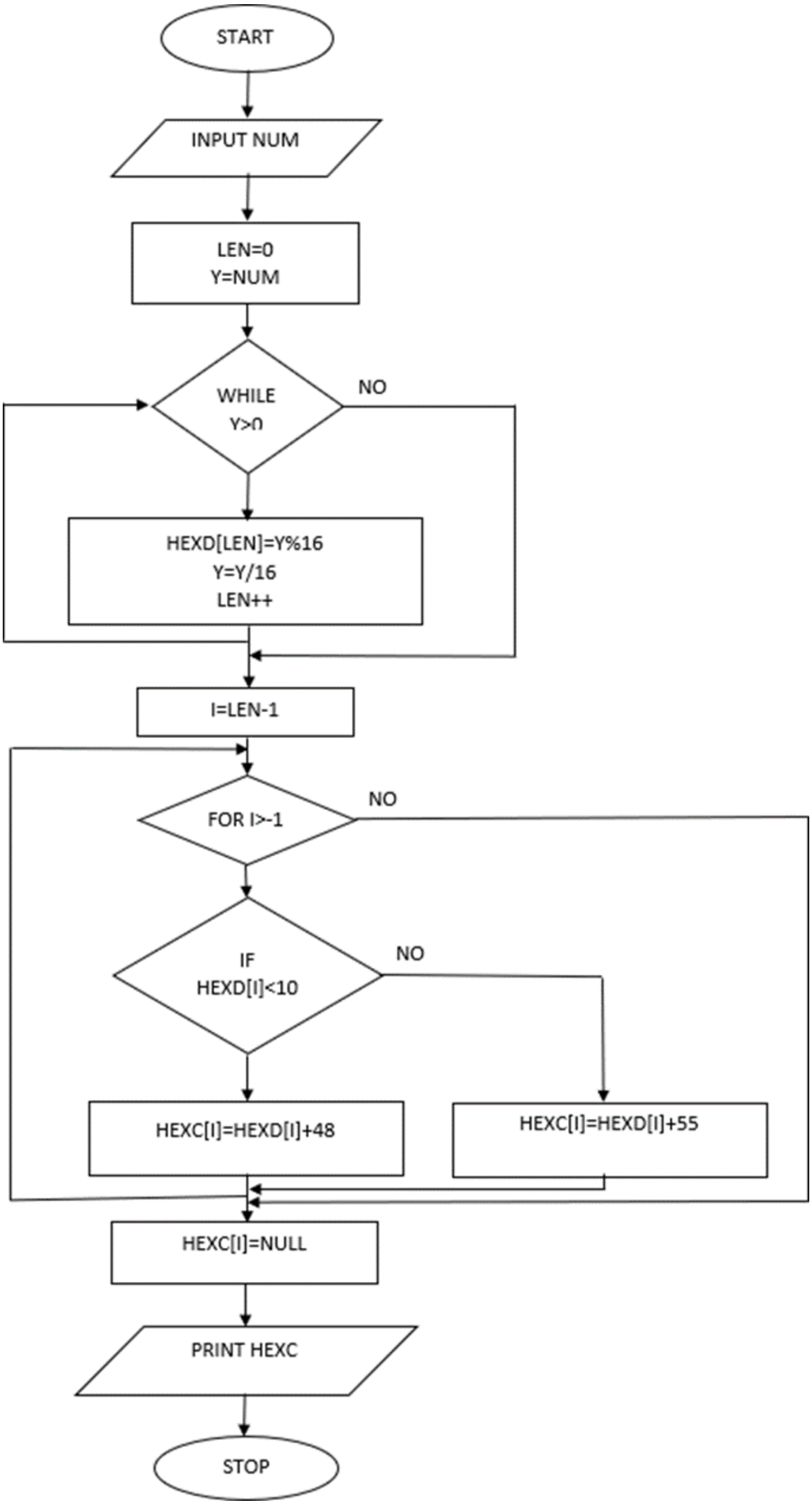
LAB ASSIGNMENTS

1. Write an algorithm, draw a corresponding flowchart and write an interactive program to convert decimal number to its hexadecimal equivalent

Algorithm

```
step 1: start
step 2: declare decimalNumber, reminder, quotient, i, j, temp,
hexadecimalNumber step 3: let i=1
step 4: read value for
decimalNumber step 5:
quotient=decimalNumber
step 6: check condition(quotient!=0),if true then go to step 7 otherwise
go to step 13 step 7: temp=quotient % 16
step 8: check condition(temp<10),if true then go to step 9 otherwise go
to step 10 step 9: temp=temp+48
step 10: temp=temp+55
step 11:
hexadecimalNumber[i++] =temp
step 12: go to step 6
step 13: quotient =
quotient/16 step 14: let
j=i-1
step 15: check condition(j>0),if true then go to step 15 otherwise go
to step 17 step 16: print decimalNumber
step 17: j–
step 18: go to step 15
step 18: stop
```

Flowchart



Program

```
  #include<stdio.
  h>
  #include<conio
  .h> void main(
  )
{
  long int decimal Number, remainder,
  quotient; int i=1, j, temp;
  char hexadecimal
  Number[100]; printf("Enter
  any decimal number: ");
  scanf("%ld",&decimalNumber)
  ; quotient = decimalNumber;
  while(quotient!=0)
{
  temp=quotient % 16;
  // To convert interger into
  character if( temp< 10)
  temp=temp
  +48; else
  temp=temp
  +55;
  hexadecimalNumber[i++]=t
  emp; quotient =
  quotient/16;
}
  printf("Equivalent Hexadecimal Value of the decimal number %d: ",
  decimalNumber); for (j=i-1 ; j>0 ; j--)
  printf("%c",
  hexadecimalNumber[j] ); getch(
  );
 }
```

Output

```
Enter any decimal number: 45
Equivalent hexadecimal value of decimal number 45: 2D
```

2. Draw a flowchart and write its corresponding C program to convert an octal number to its equivalent decimal number.

Algorithm

Step 1: Start

Step 2: Input the octal number

Step 3: Find the number of digits in the

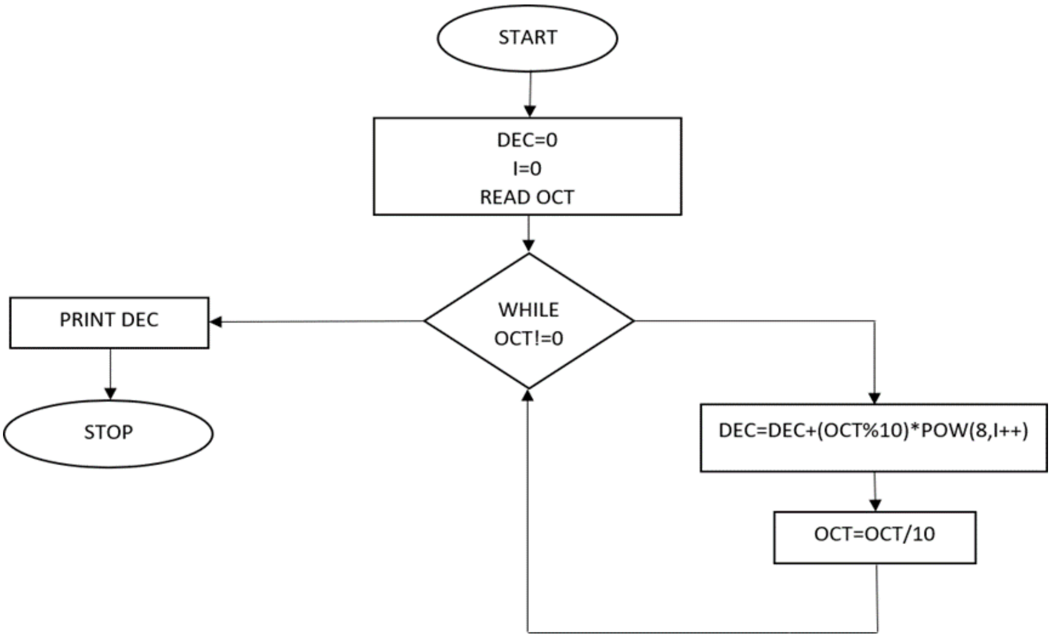
number Step 4: Let it have n digits

Step 5: Multiply each digit in the number with 8n-1, when the digit is in the

nth position Step 6: Add all digits after multiplication

Step 7: Stop

Flowchart



Program

```
#include<stdio.h>

#include<conio.h>

#include<math.h>

void main( )

{

long int

octal,decimal=0;

int i=0;

printf(“Enter any octal number: ”);
scanf(“%ld”, &octal);
while (octal != 0)

  {

  decimal = decimal+ (octal % 10)* pow(8, i++);

  octal = octal/10;

  }

printf(“Equivalent decimal value: %ld”,decimal);

getch();

      }
```

Output

```
Enter any octal number: 67

Equivalent decimal value: 55
```

3. Write an interactive C program to find the MINIMUM and MAXIMUM (value) array elements in a given 3*3 matrix

Algorithm

step 1: START

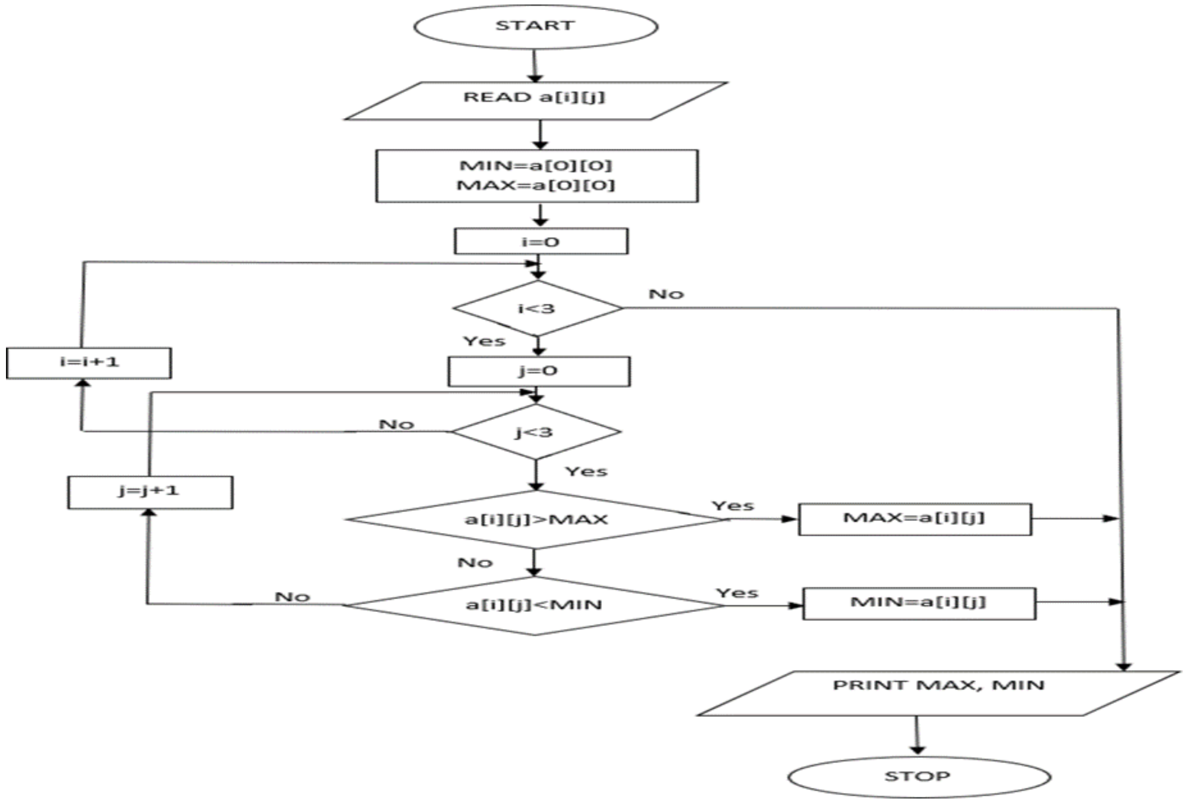
step 2: declare an array a[10][10] and variables

min,c,d,max; step 3: read array elements

step 4:

```
min=max=a[0][0]
step 5: print matrix a
step 6: intialise c=0
step 7: check FOR condition(c<3),if true then go to step 8,otherwise go to
step 14 step 8: intialise d=0
step 9: check FOR condition(d<3),if true then go to step 10,otherwise go to
step 13 step 10: check condition(a[c][d]>max),if true then go to step 11 else
go to step 12 step 11: max=a[c][d]
step 12: d++
step 13: go to step 9
step 14: c++
step 15: go to step 7
step 16: intialise c=0
step 17: check FOR condition(c<3),if true then go to step 16,otherwise go to
step 22 step 18: intialise d=0
step 19: check FOR condition(d<3),if true then go to step 18,otherwise go to
step 21 step 20: check condition(a[c][d]<min),if true then go to step 19
else go to step 20 step 21: min=a[c][d]
step 22: d++
step 23: go to step 17
step 24: c++
step 25: go to step 15
step 26: print smallest
step 27: print largest
step 28: STOP
```

Flowchart



Program

```
#include<stdio.h>
#include<conio.h>
void main( )
{
int a[10][10], min,
c, d, max;
clrscr( );
printf("Enter the
array elements");
for(c=0; c<3; c++)
{
for(d=0; d<3;
d++)
{
scanf("%d",
&a[c][d]);
min=max=a[0][0];
}
}
printf("Matrix is:
\n");
for(c=0; c<3; c++)
{
printf("\n");
for(d=0; d<3;
d++)
{
if (a[c][d] < min)
min=a[c][d];
}
}
printf("\n Smallest
element is %d",
min);
printf("\n Largest
element is %d",
max);
getch( );
}
```

Output

```
Enter the array elements 11 22 33 44 55 66 77 88
99 Matrix is:
11 22 33
44 55 66
77 88 99
```

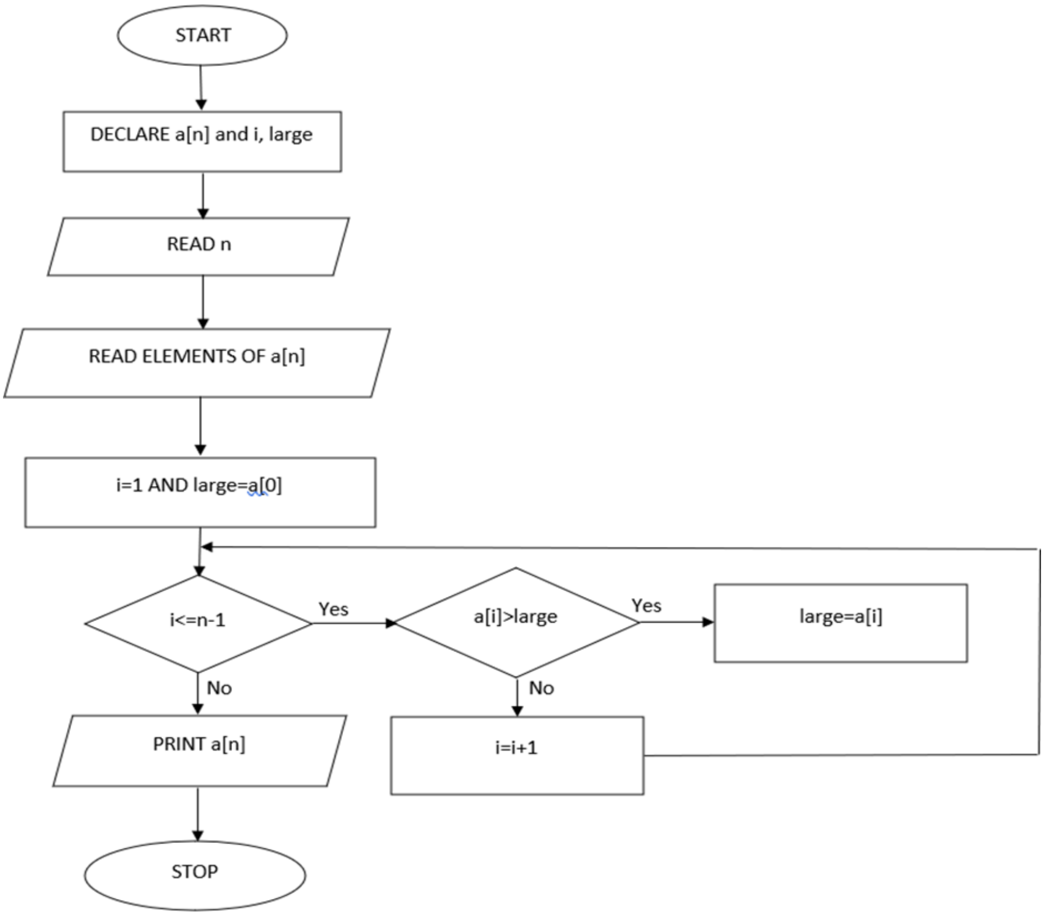
```
Smallest element is 11

Largest element is 99
```

4. Write a program to find the largest element in an array using Recursion

Algorithm

```
step 1: START
step 2: declare
size,largest,i,list[20] step 3: read
size of array
step 4: read elements of array
step 5: check condition(size==0),if true then go to step 6 otherwise go
to step 7 step 6: print list is empty
step 7: largest=list[0]
step 8: largest=large(list, size-1,
largest) step 9: print largest number
in the list step 10: STOP
int large(int list[], int position, int
largest) step 1: start
step 2: check condition(position==0),if true then go to step 2 otherwise go
to step 7 step 3: return largest
step 4: check condition(position>0),if true then go to step 6 otherwise go to step 7
step 5: check condition(list[position]>largest),if true then go to step 6 otherwise go
to step 7 step 6: largest=list[position]
step 7: return large(list,position-
1,largest
step 8: stop
```

Flowchart



Program

```
#include<stdio.h>
#include<conio.h>
int large(int[], int,int);
void main()
{
Int
size,largest,i,list[20]
; clrscr();
printf("Enter the size of the array:");
scanf("%d",&size);
printf("Enter the elements of the
array:"); for(i=0;i<size;i++)
{
```

```
scanf("%d",&list[i]);
}
if(size==0)
{
printf("Empty list \n");
}
else
{
list[0];
largest=large(list, size-1, largest);
printf("\n The largest number in the list is: %d \n",largest);
}
getch();
}
int large(int list[], int position, int largest)
{
if(position==0
) return
largest;
if(position>0)
{
if(list[position]>largest)
{
largest=list[position];
}
return large(list,position-1,largest);
}
}
```

Output

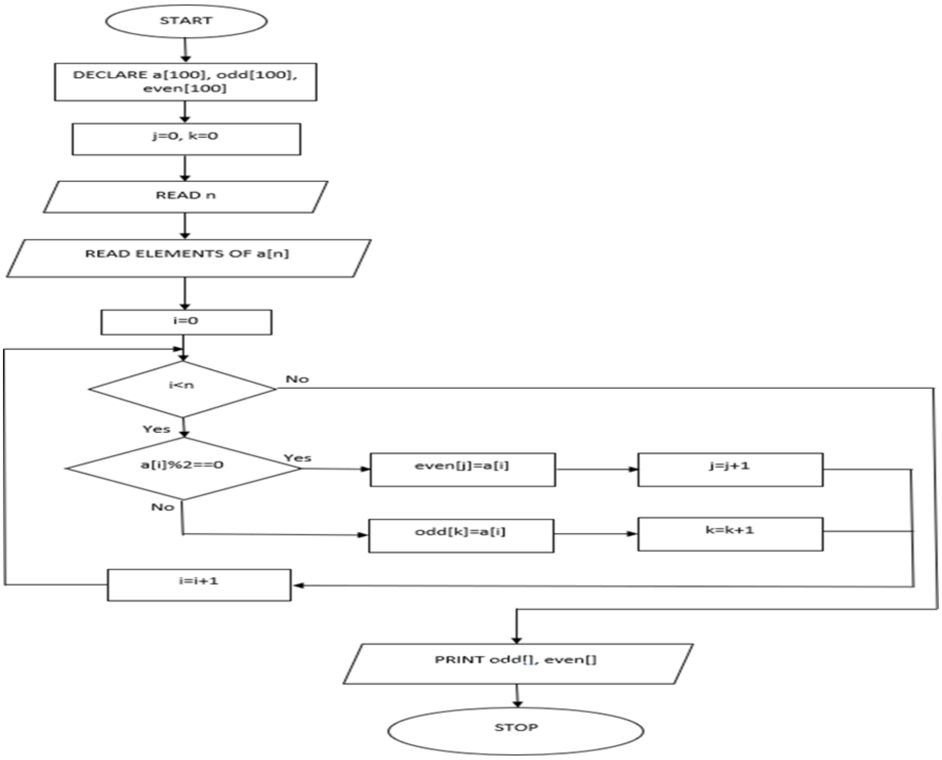
```
Enter the size of the array: 5
Enter the elements of the array: 11 22 55
88 44 The largest number in the list is: 88
```

5. Write a C program to separate even and odd numbers of an array and put them in two separate arrays.

Algorithm

```
Step 1: Start
Step 2: Declare arrays a[100], odd[100], even[100] and variables
n,i,j=0,k=0 Step 3: Input size of the array a
Step 4: Read the array
elements Step 5: intialisei=0
Step 6: check condition(i<n),if true then go to step 7 otherwise go to
step 14 Step 7: check condition(a[i]%2==0),if true go to step 8
otherwise go to step 10 Step 8: even[j]=a[i]
Step 9: j++
Step 10: odd[k]=a[i]
Step 11: k++
Step 12: i++
Step 13: go to step 6
Step 14: intialisei=0
Step 15: check condition(i<k),if true then go to step 16 otherwise go to
step 19 Step 16: print odd numbers
Step 17: i++
Step 18: go to step 15
Step 19: intialisei=0
Step 20: check condition(i<j),if true then go to step 21 otherwise go to
step 24 Step 21: print even numbers
Step 22: i++
Step 23: go to step 20
Step 24: Stop
```

Flowchart



Program

```
#include<stdio.h>
#include<conio.h>
void main( )
{
int
a[20],odd[20],even[20],i,j=0,k=0,n;
clrscr();
printf("Enter the size of an
array"); scanf("%d",&n);
printf("Enter the array
elements"); for(i=0;i<n;i++)
{
 scanf("%d",&a[i]);
}
for(i=0;i<n;i++)
{
if(a[i]%2==0)
{
even[j]=a
[i]; j++;
}
Else
{
odd[k]=a[
i]; k++;
}
}
printf("The elements of odd numbers in the array\n");
for(i=0;i<k;i++)
{
printf("%d\t",odd[i]);
}
printf("\nThe elements of even numbers in the
array\n"); for(i=0;i<j;i++)
{
printf("%d\t",even[i]);
}
getch();
}
```

***Outpu*t**

```
Enter the size of an array 5
Enter the array elements 1 2 3 4 5
The elements of the odd numbers in the
```

```
array 1 3 5
The elements of the even numbers in the array
 1   4
```

6. Write the following functions that:

a. Calculate simple interest

b. Calculate compound interest

Write an interactive C (main) program to provide the above functions as options to the user using switch statement and performs the functions accordingly.

Algorithm

```
Step 1: Start
Step 2: Read Principal, Rate and Time
Step 3: display to press 1 for simple interest and 2 for compound
interest Step 4: read choice
Step 5: If choice=1
Then
SI=((principal*rate*time)/100)
Print SI
Step 6: If choice=2
Then CI=Principal*((1+rate/100)
time) Print CI
Step 7: Stop
```

Program

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

```
#include<math.
h> void main( )
 {
    float
    p,r,t,si,ci,
    a; int ch;
    clrscr();
    printf("Enter the principal amount:");
    scanf("%f",&p);
    printf("Enter the rate of interest:");
    scanf("%f",&r);
    printf("Enter the
    time:");
    scanf("%f",&t);
    printf("\n 1.For Simple
    Interest"); printf("\n 2.For
    Compound Interest"); printf("\n
    Select any option");
    scanf("%d",&ch);
    switch(ch)
 {
    case 1:
    si=(p*r*t)/100;
    printf("Simple interest is
    %f",si); break;
    case 2:
    ci=p*pow((1+r/10
    0),t)-p;
    printf("Compound interest is
    %f",ci); break;
    default:
```

```
    printf("Invalid

    choice"); break;

  }

    getch();

  }
```

Output

```
Enter the principal amount:

2000 Enter the rate of interest:

3

Enter the time: 2

1. For simple interest

2. For compound interest

Select any option 1Simple interest is

120.000000 Enter the principal amount:

5400

Enter the rate of interest: 8
Enter the time: 3

The compound interest is 1402.444824
```

7. Write an algorithm and its corresponding C program to illustrate an ATM money withdrawal operation from user's savings' account.

Algorithm

```
Step 1: Start

Step 2: Declare variables amount=15000, deposit, withdraw, choice, pin, k, try=0,

transaction=y Step 3: Read value pin

Step 4: If pin != 4567 then print "You have entered invalid pin

number" Step 5: Increment value of try

Step 6: If try == 3 then print "You have reached the maximum attempt to enter pin

number" Step 7: Exit

Step 8: do
```

```
  Step 9: print “Welcome to YTH Bank
1. Check balance
2. Withdraw cash
3. Deposit cash
4. Quit
  10: Read choice
  Step 11: Apply switch case to select the
  operator Case 1: Print the balance amount
  Case 2: Read withdraw
  If withdraw != 0 then print please enter  in multiples of
  100 Else if withdraw>(amount-1000) then print
  insufficient balance Else set amount = amount-withdraw
  Print please collect your
  cash Print amount
  Case 3: Read deposit
  Set amount=amount +
  deposit Print amount
  Case 4: print Thank you for using this ATM
  Default: Print invalid option
  Step 12: Continue step 11 until transaction == n
  or N Step 13: Stop
```

Program

```
#include<stdio.h>
#include<conio.h>
unsigned long amount=15000, deposit,
withdraw; int choice, pin, k, TRY=0;
char
transaction='Y';
void main()
```

```
{
clrscr();
while(pin!=4567)
{
printf("Enter your pin
number"); scanf("%d",&pin);
if(pin!=4567)
{
printf("\n You have entered invalid pin
number"); TRY++;
if(TRY==3)
{
printf("You have reached the maximum attempt to enter pin
number!!\n"); exit();
}
}
}
do
{clrscr();
printf("Welcome to YTH Bank\n");
printf("1. Check
balance\n");
printf("2.Withdraw
cash\n");
printf("3.Deposit
cash\n");
printf("4.Quit\n");
printf("Enter your
choice");
scanf("%d",&choice);
switch(choice)
```

```
{
case 1:
printf("Your balance is Rs.%d
\n",amount); break;
case 2:
printf("Enter your amount to withdraw cash\n");
scanf("%ld",&withdraw);
if(withdraw%100!=0)
{
printf("Please enter the amount in multiples of 100\n");
}
else if(withdraw>(amount-1000))
{
printf("\n \nSufficient balance\n");
}
else
{
amount=amount-withdraw;
printf("Please collect your
cash\n");
printf("Your current balance is %ld \n",amount);
}break; case 3:
printf("\n Enter the amount to deposit");
scanf("%ld",&deposit);
amount=amount+deposit;
printf("Your current balance is %ld
\n",amount); break;
case 4:
printf("Thank you for using this ATM
\n"); break;
default:
```

```
printf("Invalid option");
printf("Do you wish to do another transaction");
fflush(stdin);
scanf("%c",&transaction);
if(transaction=='n'||transaction=='N')
{
K=1;
}
Else
K=0;}while(!k)
getch();
}
```

Output

```
Enter your pin number 4567
Welcome to YTH Bank
 1. Check balance
 2. Withdraw cash
 3. Deposit cash
 4. Quit
   Enter your choice 1
   Your balance is Rs.15000
   Do you wish to do another
   transaction y Welcome to YTH Bank
 1. Check balance
 2. Withdraw cash
 3. Deposit cash
 4. Quit
   Enter your choice 2
   Enter your amount to
   withdraw cash 1000
   Please collect your cash
   Your current balance is 14000
   Do you wish to do another
   transaction y Welcome to YTH Bank


 1. Check balance
 2. Withdraw cash
 3. Deposit cash
 4. Quit
   Enter your choice 3
   Enter the amount to deposit
   2000 Your current balance
   is 16000
   Do you wish to do another
   transaction y Welcome to YTH Bank
 1. Check balance
 2. Withdraw cash
 3. Deposit cash
 4. Quit
   Enter your choice 4
   Thank you for using this ATM
   Do you wish to do another transaction n
```

8. Write the following string functions that:

i. Replace a character in a given string with a character suggested by the user.

ii. Convert the given string into uppercase.

iii. Convert the alternate character in the string into uppercase.

iv. Check each and every character in the string and display whether it is an alphabet digit or special character.

Write an interactive C(main) program to provide the above string functions as options to the user using switch statement and perform the functions accordingly.

Algorithm

```
Step 1: Start
Step 2: declare
ch,k=0,choice Step 3: do
tep 4: print 1. Replace the
character Step 5: print 2.
Convert uppercase
Step 6: print 3. Convert alternate character in
uppercase Step 7: print 4. Check alphabet, digit,
special character Step 8: take input for choice
Step 9: case 1: print Replace the
character call replace()
case 2: print Convert
uppercase call uppercase()
case 3:print Convert alternate character into
uppercase call alteruppercase()
case 4:print Check alphabet, digit, special character in a
string call check()
default:print Invalid
choice Step 10: read
value for ch
Step 11: check condition(ch=='n'||ch=='N'),if true go to step 12 otherwise go
to 13 Step 12: k=1
Step 13: k=0
Step 14: check condition(k!=1),if true go to step 3 otherwise go
to 15 Step 15: Stop
```

replace()

Step 1: Start

Step 2: declare array str[100] and variables

chr,newchr,i,index=-1 Step 3: read value for str

Step 4: read character to

replace Step 5: read new

character

Step 6: initialize i=0

Step 7: check condition(str[i]!='\0'),if true go to step 8 otherwise go to

step 12 Step 8: check condition(str[i]==chr),if true go to step 9

otherwise go to step 11 Step 9: index=i

Step 10: i++

Step 11: go to step 7

Step 12: check condition(index!=-1),if true go to step 13 otherwise go to

step 14 Step 13: str[index]=newchr

Step 14: print the final string after replacing

variable Step 15: Stop

uppercase()

Step 1: Start

Step 2: declare array str1[100] and

variable i Step 3: read value for string

Step 4: initialize i=0

Step 5: check condition(str[i]!='\0'),if true go to step 6 otherwise go to step 10

Step 6: check condition(str1[i]>='a'&&str1[i]<='z'),if true go to step 7 otherwise go

to step 8 Step 7: str1[i]=str1[i]-32

Step 8: i++

Step 9: go to step 5

Step 10: print string in

uppercase Step 11: Stop

```
alteruppercase()
Step 1: Start
Step 2: declare array str2[100] and variable i
Step 3: read string
Step 4: initialize i=0
Step 5: check condition(str[i]!='\0'),if true go to step 6 otherwise go to step 13
Step 6: check condition(i%2==0),if true go to step 7 otherwise go to step 9
Step 7: check condition(str2[i]>='A'&&str2[i]<='Z'),if true go to step 8 otherwise go to step 9
Step 8: str2[i]=str2[i]+32
Step 9: check condition(str2[i]>='a'&&str2[i]<='z),if true go to step 10 otherwise go to step 11
Step 10: str2[i]=str2[i]-32
Step 11: i++
Step 12: go to step 5
Step 13: print the alternate converted string
Step 14: Stop
```

```
check()
Step 1: Start
Step 2: declare array str3[100] and variable i=0
Step 3: read value of string
Step 4: check condition(str3[i]!='\0'),if true go to step 5 otherwise go to step 12
Step 5: check condition((str3[i]>'a'&&str3[i]<='z')||(str3[i]>'A'&&str3[i]<='Z')),if true go to step 6 otherwise go to step 7
Step 6: print character is an alphabet
Step 7: check condition(str3[i]>='0'&&str3[i]<='9'),if true go to step 8 otherwise go to step 9
Step 8: print character is a digit
Step 9: print character is a special character
Step 10: i++
Step 11: go to step 4
Step 12: stop
```

Program

```
#include<stdio.

h>

#include<conio.

h> void

replace(); void

uppercase();

void

alteruppercase();

void check();

void main()
{

char ch;

int k=0,choice;

clrscr();

do

{

printf("\n 1. Replace the

character"); printf("\n 2. Convert

uppercase");

printf("\n 3. Convert alternate character in

uppercase"); printf("\n 4. Check alphabet, digit,

special character"); printf("\n Enter the choice:");

scanf("%d",&choice);

switch(choice)


{

case 1:

printf("\n Replace the

character"); replace();break;
case 2:
printf("\n Convert
uppercase");
uppercase();break; case 3:
printf("\n Convert alternate character into
uppercase"); alteruppercase();brake;case4:
printf("\n Check alphabet, digit, special character in a
string"); check();
break;
default:pr
intf("\n
Invalid
choice");
break;
}
printf("\n DO YOU WANT TO CONTINUE");
fflush(stdin);
scanf("%c",&ch);
if(ch=='n'||ch=='N')
{k=1;}}while(!k); getch();
}
void replace()
{
char
str[100],chr,newchr;
int i,index;
index=-1;
printf("\n Enter the string:");
scanf("%s",&str);
printf("\n Enter the character that you want to
replace:"); fflush(stdin);
```

```
scanf("%c",&chr);
printf("\n Please enter the new
character:"); fflush(stdin);
scanf("%c",&newc
hr);
for(i=0;str[i]!='\0';i
++)
{
if(str[i]==chr)
{
index=i;
}}
if(index!=-1)
{
str[index]=newchr;
}
printf("\n The final string after replacing last occurance of %c with %c=%s",chr,newchr,str);
}
void uppercase()
{
char
str1[100];
int i;
printf("\n Enter the string: ");
scanf("%s",&str1);
for(i=0;str1[i]!='\0';i++)
{
if(str1[i]>='a'&&str1[i]<='z')
{
str1[i]=str1[i]-32;
}}
```

```
printf("\n The given string in uppercase: %s",str1);
}voidalteruppercase()
{
char
str2[100];
int i;
printf("\n Enter the string:");
scanf("%s",&str2);
for(i=0;str2[i]!='\0';i++){if((i%
2)==0)
{
if(str2[i]>='A'&&str2[i]<='Z')
{
str2[i]=str2[i]+32;
}}
else
{
if(str2[i]>='a'&&str2[i]<='z')
{
str2[i]=str2[i]-32;
}
}}
printf("\n The alternate converted string is %s",str2);
}
void check()
{
char
str3[100];
int i=0;
printf("\n Enter the string:");
```

```
scanf("%s",&str3);

while(str3[i]!='\0')

{

if((str3[i]>'a'&&str3[i]<='z')||(str3[i]>'A'&&str3[i]<='Z'))

{

printf("\n %c is a alphabet",str3[i]);

}


else if(str3[i]>='0'&&str3[i]<='9')

{

printf("\n %c is a digit",str3[i]);

}

else

{

printf("\n %c is a special character",str3[i]);} i++;
}}
```

Output

```
 1. Replace the character

 2.Convert uppercase

 3.Convert alternate character in uppercase

        1. Check alphabet, digit, special

           character Enter the choice: 1

    Replace the

    character Enter the

    string: james

    Enter the character that you want to

    replace: e Please enter the new

    character: i

    The final string after replacing the last occurance of e

    with i: jamis DO YOU WANT TO CONTINUE ? Y
```

```
1. Replace the character
2. Convert uppercase
3. Convert alternate character in uppercase
4. Check alphabet, digit, special
character Enter the choice: 2
   Convert uppercase
   Enter the string: welcome
   The given string in uppercase:
   WELCOME DO YOU WANT TO
   CONTINUE? Y
1. Replace the character
2. Convert uppercase
3. Convert alternate character in uppercase
4. Check alphabet, digit, special
character


5. Enter the choice: 3
   Convert alternate character into
   uppercase Enter the string: world
   The alternate converted string is
   world DO YOU WANT TO
   CONTINUE? Y
1. Replace the character
2. Convert uppercase
3. Convert alternate character in uppercase
4. Check alphabet, digit, special
character Enter the choice: 4
   Check alphabet, digit, special character in
   a string Enter a string: im@3000
   i is a
   alphabet
   m is a
   alphabet
@ is a special
character 3 is a
digit 0 is a digit 0
is a digit 0 is a
digit
   DO YOU WANT TO CONTINUE ? N
```

9. Write a program to search a given string among the available strings, using Binary Search.

Algorithm

```
Step 1: Start
Step 2: declare arrays str[20][50],s1[50] and variables
i,n,found=0 Step 3: read limit
Step 4: read string
value Step 5: initialize
i=0
Step 6: check condition(i<n),if true go to step 7 otherwise go to step 12
Step 7: check condition(strcmp(s1,str[i])==0),if true go to step 8 otherwise go
to step 10 Step 8: found=1
Step 9: print found in
row Step 10: i++
Step 11: go to step 6
Step 12: check condition(found==0),if true go to step 13 otherwise go to
step 14 Step 13: print not found
Step 14: Stop
```

Program

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

```
#include<conio
.h>
#include<string
.h> void main()
{
char str[20][50],
s1[50]; int i, n,
found=0; clrscr();
printf("Enter how many string (names):
"); scanf("%d",&n);
printf("Enter %d
strings:\n",n); for(i=0;i<n;i++)
{
scanf("%s",str[i]);
}
printf("Enter a string to search:
"); scanf("%s",s1);
for(i=0;i<n;i++)
{
if(strcmp(s1,str[i])==0)
{
found=1;
printf("Found in row - %d\n",i+1);
}
}
if(found==0)
printf("Not
found"); getch();
}
```

Output

```
Enter how many strings

(names): 3

Ammu

    Anu
    Akku

    Enter a string to search:

    Anu Found in row – 2
```

10. Using structures concept in C programming, write a program to calculate the daily wages for each worker (if 7 workers are employed in an iron and hardware shop) at an hourly basis of Rs.100/- (with a constraint that each worker may be allowed maximum up to 4 hours only per day). It should display the name of the worker, date and total wages for that day.

Algorithm

```
Step 1: Start

Step 2: Declare structure worker with variables

name,hours,wages Step 3: Declare variables i,date

Step 4: copy names to structure variable

name Step 5: read date

Step 6: initialize i=0

Step 7: check condition(i<7),if true go to step 8 otherwise go to

step 16 Step 8: do

Step 9: read total working hours of each workers

Step 10: check condition(w[i].hours<= 4),if true go to step 11 otherwise go to

step 12 Step 11: w[i].wages = w[i].hours * 100

Step 12: print wrong input

Step 13: check condition(w[i].hours> 4),if true go to step 8 otherwise go to

step 14 Step 14: i++
Step 15: go to step
7 Step 16: print
date
Step 17: print name and
wages Step 18: Stop
```

Program

```
#include<stdio.h>
#include<string.h>
/* Structure Declaration */
struct worker
{
char
name[25];
int hours;
int wages;
};
void main()
{
int i;
char date[12];
/* Structure Array Declaration
*/ struct worker w[7];
/* Declared 7 Workers name who worked in an Iron and Hardware
Shop */ strcpy(w[0].name, "AKASH");
strcpy(w[1].name, "ROHIT");
strcpy(w[2].name, "RAJU");
strcpy(w[3].name, "JAGGU");
strcpy(w[4].name, "CHUTKKI");
strcpy(w[5].name, "BHEEM");
w[6].name, "KAALIYA");
```

```
printf("\n Enter the Date [DD-MM-YYYY] : ");
gets(date);
/* Input Total Working Hours for each Worker */
for(i=0;i<7;i++)
{
do
{
printf("\n Enter the Total Working Hours of %s : ",
w[i].name); scanf("%d",&w[i].hours);
if(w[i].hours<= 4)
{
w[i].wages = w[i].hours * 100;
}
else
{
printf("\n Wrong Input - Allow maximum upto 4 Hours per day for each Worker");
}}
while(w[i].hours> 4);
}
/* Display the Wages pay to each
Worker */ printf("\n\n\n Date - %s",
date); for(i=0;i<7;i++)
{
printf("\n\n Name - %s", w[i].name);
printf("\n Wages - %d", w[i].wages);
}
getch();
}
```

Output

```
Enter the Date [DD-MM-YYYY] : 12-1-2021
Enter the Total Working Hours of AKASH
```

```
: 1 Enter the Total Working Hours of
ROHIT : 2 Enter the Total Working Hours
of RAJU : 3 Enter the Total Working
Hours of JAGGU : 4 Enter the Total
Working Hours of CHUTKKI : 5
Wrong Input - Allow maximum upto 4 Hours per day for each
Workers Enter the Total Working Hours of CHUTKKI : 1
Enter the Total Working Hours of
BHEEM : 2

Enter the Total Working Hours of
KALIYA : 3 Date:-12-1-2021
Name:
AKASH
Wages - 100
Name:
RAJU
Wages - 200
Name: JAGGU
Wages - 300
Name:
CHUTKKI
Wages - 100
Name:
BHEEM
Wages - 200
Name:
KALIYA
Wages – 300
```

11. Using pointers, find the sum of all the elements of a 3*3 matrix.

Algorithm

```
Step 1:Start
Step 2:declare variables i,j,sum=0,array matrix[3][3],
pointer p Step 3:read array
Step
4:p=&matrix[0][0]
Step 5:initializei=0
Step 6:check condition(i<3),if true then go to 6 step otherwise go to
step 14 Step 7:initialize j=0
Step 8:check condition(j<3),if true then go to 9 step otherwise go to
step 12 Step 9:sum = sum + *(p + i*3 + j)
Step 10:j++
Step 11:go to step
8 Step 12:i++
Step 13:go to
step 6
Step14:print sum
Step 15:Stop
```

Program

```
#include<stdio.h
>
#include<conio.h
> void main()
{
  int matrix[3][3], i, j, sum =
  0, *p; for(i=0;i<3;i++)
{
  for(j=0;j<3;j++)
{
  printf("\n Enter the Value of Matrix[%d][%d] : ", i,
  j); scanf("%d",&matrix[i][j]);
}
}
  p =
  &matrix[0][0];
  for(i=0;i<3;i++)
{
  for(j=0;j<3;j++)
{
  sum = sum + *(p + i*3 + j);
}
}
  printf("\n Answer = %d",
  sum); getch();
}
```

Output

```
C:\TURBOC3\BIN>TC Enter the Value of
Matrix[0][0] : 1 Enter the Value of Matrix[0][1]:2
Enter the Value of
Matrix[0][2] : 3

Enter the Value of
Matrix[1][0] : 4
Enter the Value of
Matrix[1][1] : 5

Enter the Value of
```

```
Matrix[1][2] : 6 Enter the
Value of Matrix[2][0] : 7
Enter the Value of
Matrix[2][1] : 8 Enter the
Value of Matrix[2][2] : 9
Answer = 45
```

12. Write an interactive C program to calculate the sum of array elements using pointer.

Algorithm

Step 1: Start

Step 2: Declare variables i,n,sum=0 and

pointer a Step 3: Read size of array

Step 4: Dynamically allocate memory for

pointer a Step 5: Read elements of list

Step 6: Initialize variable i=0

Step 7: Check condition(i<n),if true then go to step 8 otherwise go to

step 11 Step 8: sum = sum + *(a+i)

Step 9: i++

Step 10: Go to

step 7 Step 11:

Print sum Step 12:

Stop

Program

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

```
void main()
{
int i, n,
sum=0; int
*a;
printf("Enter thr size of the array A \n");
scanf("%d", &n);
a = (int *) malloc(n * sizeof(int));
printf("Enter elements of the list \n");
for(i=0;i<n;i++)
{
scanf("%d", a+i);
}
for(i=0;i<n;i++)
{
sum = sum + *(a+i);
}
printf("Sum of all the elements in array = %d\n",
sum); getch();
}
```

Output

```
Enter the size of the
array A 5
Enter elements of the
list 1
2
3
4
5
Sum of all elements in array = 15
```

13. Using file handling, write a C program

i. To generate 10 records for MCA 1st semester students and store them in stu.dat along with appropriate fields.

Algorithm

Step 1: Start

Step 2: declare structure result with variable subject_code,marks and structure student with variable enrolment_no,name

Step 3: declare variable i

Step 4: copy respective subject codes to strucutre variable

array

Step 5: open file to write

Step 6: check condition(stream==NULL),if true then go to step 7 otherwise go

to step 8

Step 7: print cannot open file

Step 8: initialize variable i=1

Step 9: check for condition(i<=1),if true then go to step 10 otherwise go to

step 12 Step 10:read details of student

Step 11:write it to

file 12:Stop

Program

```
#include<stdio.

h>

#include<string.

h>

#include<conio

.h> struct result

{

char
```

```
subject_code[10];
int marks;
};
struct student
{
char enrolment_no[10];
char name[25];
struct result sem_I[7];
};
void main()
{
int i; FILE *stream;
struct student s;
strcpy(s.sem_I[0].subject_code, "MCS-
011"); strcpy(s.sem_I[1].subject_code,
"MCS-012");
strcpy(s.sem_I[2].subject_code, "MCS-
013");
strcpy(s.sem_I[3].subject_code, "MCS-
014"); strcpy(s.sem_I[4].subject_code,
"MCS-015");
strcpy(s.sem_I[5].subject_code, "MCSL-
016"); strcpy(s.sem_I[6].subject_code,
"MCSL-017"); stream =
fopen("stu.dat","w");
if(stream == NULL)
{
printf("\nError! Can not open output file");
}
else
{
```

```
for(i=1;i<=10;i++)
{
printf("\nEnter the Student Enrolment Number : ");
fflush(stdin);
gets(s.enrolment_no);
printf("\nEnter the Student Name : ");
fflush(stdin);
gets(s.name);
printf("\nEnter the Marks obtain in MCS-011 : ");
scanf("%d",&s.sem_I[0].marks);
printf("Enter the Marks obtain in MCS-012 : ");
scanf("%d",&s.sem_I[1].marks);
printf("Enter the Marks obtain in MCS-013 : ");
scanf("%d",&s.sem_I[2].marks);
printf("Enter the Marks obtain in MCS-014 : ");
scanf("%d",&s.sem_I[3].marks);
printf("Enter the Marks obtain in MCS-015 : ");
scanf("%d",&s.sem_I[4].marks);
printf("Enter the Marks obtain in MCSL-016 : ");
scanf("%d",&s.sem_I[5].marks);
printf("Enter the Marks obtain in MCSL-017 : ");
scanf("%d",&s.sem_I[6].marks);
fwrite(&s, sizeof(s), 1, stream);
}
fclose(stream);
}
getch();
}
```

Output

```
Enter the Student Enrolment Number :
119 Enter the Student Name :donajose
Enter the Marks obtain in MCS-011 :
80 Enter the Marks obtain in MCS-
012 : 80 Enter the Marks obtain in
MCS-013 : 80 Enter the Marks obtain
in MCS-014 : 80 Enter the Marks
obtain in MCS-015 : 80 Enter the
Marks obtain in MCSL-016 : 80
Enter the Marks obtain in MCSL-017
: 80 Enter the Student Enrolment
Number : 120 Enter the Student
Name :divya joseph Enter the Marks
obtain in MCS-011 : 90 Enter the
Marks obtain in MCS-012 : 90 Enter
the Marks obtain in MCS-013 : 80
Enter the Marks obtain in MCS-014 :
90 Enter the Marks obtain in MCS-
015 : 80 Enter the Marks obtain in
MCSL-016 : 90 Enter the Marks
obtain in MCSL-017 : 80
```

14. To read the data from the file stu.dat (created above) and compute the total marks and average marks and display the grade assumptions can be made).

Algorithm

```
Step 1: Start
  Step 2: declare structure variable
  subject_code,marks,enrolment_no,name Step 3: declare variables
  i,total_mark,average_marks
```

```
Step 4: open file to read
Step 5: check condition(stream==NULL),if true then go to step 6 otherwise go to step 7
Step 6: print cannot open file
Step 7: check condition(fread(&s, sizeof(s), 1, stream)),if true then go to step 8 otherwise go to step 24
Step 8: print name and enrollment_no
Step 9: total_marks = 0
Step 10:initialize variable i=0
Step 11:check condition(stream==NULL),if true then go to step 12 otherwise go to step 15
Step 12:total_marks = total_marks + s.sem_I[i].marks
Step 13:i++
Step 14:go to step 11
Step 15:average_marks = total_marks / 7
Step 16:print total mark and average mark
Step 17:check condition(average_marks>= 40 &&average_marks< 60),if true then go to step 18 otherwise go to step 19
Step 18:print grade – D
Step 19:check condition(average_marks>= 60 &&average_marks< 70),if true then go to step 20 otherwise go to step 21
Step 20:print grade - C
Step 21:check condition(average_marks>= 70 &&average_marks< 80),if true then go to step 22 otherwise go to step 23
Step 22:print grade - B
Step 23: go to step 7
Step 24: Stop
```

Program

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

```
struct result
{
char
subject_code[10];
int marks;
};
struct student
{
char enrolment_no[10];
char name[25];
struct result sem_I[7];
};
void main()
{
FILE
*stream;
struct
student s;
int i, total_marks,
average_marks; stream =
fopen("D:\\stu.dat","r");
if(stream == NULL)
{
printf("\nError! Can not open output file");
}
else#
{
while(fread(&s, sizeof(s), 1, stream))
{
printf("\n\n Enrolment No - %s",
s.enrolment_no); printf("\n Name - %s",
s.name);
total_marks = 0;
for(i=0;i<7;i++)
{
total_marks = total_marks + s.sem_I[i].marks;
}
average_marks  =  total_marks  /  7;

printf("\n Total Marks - %d",
total_marks);
printf("\n Average Marks - %d",
average_marks); if(average_marks>= 40
&&average_marks< 60) printf("\n Grade - D");
else
if(average_marks>= 60 &&average_marks< 70)
printf("\n Grade - C");
else
if(average_marks>= 70 &&average_marks< 80)
printf("\n Grade - B");
else
if(average_marks>= 80 &&average_marks< 90)
printf("\n Grade - A");
else
if(average_marks>= 90 &&average_marks<=
100) printf("\n Grade - A+");
}
fclose(stream);
}
getch();
}
```

Output

```
Grade - B
Enrolment No -
117
Name - anjosemathew
Total Marks - 560
Average Marks - 80

Grade - A
Enrolment No -
118 Name -
treesa joseph
Total Marks -
630 Average
Marks - 90

Grade - A+
Enrolment No -
119 Name -
donajose Total
Marks - 560
Average Marks -
80 Grade - A
Enrolment No -
120 Name -
divya joseph
Total Marks -
600 Average
Marks - 85
Grade – A
```

15. Write an interactive C program to append the contents of a file at the end of another file without using any built-in functions.

Algorithm

```
Step 1: Start
Step 2: Declare fname1,fname2,c
Step 3: read a filename to open for
reading Step 4: fp1 = fopen(fname1, "r")
Step 5: check condition(fp1==NULL),if true then go to step 6 otherwise go
to step 7 Step 6: print file does not exist
Step 7: read filename to append the
content Step 8: fp2=fopen(fname2,"a")
Step 9: check condition(fp2==NULL),if true then go to step 10 otherwise go to
step 11 Step 10: print file does not exist
Step 11: c = fgetc(fp1)
Step 12: check condition(c!=EOF),if true then go to step 13 otherwise go to
step 11
Step 13: write content of first file to second file
Step 14: print content in first file appended to second
file Step 15: Stop
```

Program

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


#include<stdlib.h>
void main()
{
FILE *fp1, *fp2;
char fname1[50],
fname2[50],c; clrscr();
printf("Enter the filename to open for reading: ");
scanf("%s", fname1);
fp1 = fopen(fname1, "r");
if(fp1==NULL)
{
printf("%s file does not exist..",fname1);
getch();
}
printf("\n Enter filename to append the content: ");
scanf("%s",fname2);
fp2=fopen(fname2,"a");
if(fp2==NULL)
{
printf("%s file does not exist..",fname2);
getch();
}
c = fgetc(fp1);
while(c!=EOF)
{
fputc(c,fp2);
c=fgetc(fp1);
}
printf("\n Content in %s appended to
%s",fname1,fname2); fclose(fp1);
fclose(fp2
); getch();
}
```

Output

```
Enter the filename  to open for reading:

first.txt Enter the filename to append the

content: second.txt Content in first.txt

appended to second.txt

first.txt

Welcome

second.txt

Hello,

After appending the content of first.txt, second.txt will be second.txt

Hello, Welcome
```

16. Write an interactive C program to create a file containing student’s records and also give provision to update/modify the records too.

Algorithm

```
Step 1: Start

Step 2: Declare a structure student with variables
rollno,name,test_score1,test_score2,test_score3,average,grade

Step 3: declare variables

choice Step 4: do

Step 5: print main menu

Step 6: print result menu


Step 7: print entry/edit

menu Step 8: print exit

Step 9: read value for choice and check the

cases

Step 10: case 1:callresult_menu function

case 2: call entry_menu function

Step 11:check condition(choice !=3),if true then go to step 4 otherwise go to
```

```
step 12 Step 12:Stop
char calculate_grade(float average)
Step 1: Start
Step 2: check condition(average >= 90),if true then go to step 3 otherwise go to
step 4 Step 3: return value'A'
Step 4: check condition(average >= 70),if true then go to step 5 otherwise go to
step 6 Step 5: return value 'B'
Step 6: check condition(average >= 50),if true then go to step 7 otherwise go to
step 8 Step 7: return value 'C'
Step 8: return value 'F'
Step 9: Stop
void input(struct student
*s) Step 1: Start
Step 2: Read roll number,name and marks of student
Step 3: s->average = (s->test_score1 + s->test_score2 + s-
>test_score3) / 3.0 Step 4: s->grade = calculate_grade(s->average)
Step 5: Stop
void display(struct student st)
Step 1: Start
Step 2: print roll number,name and marks of
student Step 3: print average score and grade
Step 4: Stop
void write_student()
Step 1: Start
Step 2: open student.dat file in append
mode Step 3: print enter details of
student
Step 4: call function
input Step 5: write
details to file
Step 6: print student record has been
```

created

Step 7: Stop

void display_all()

Step 1: Start

Step 2: open student.dat file in read mode

Step 3: check condition((fread(&data, sizeof(data), 1, file)) > 0),if true then go to step 4 otherwise go to step 5

Step 4: display details

Step 5: go to step 3

Step 6: Stop

void display_sp(int n)

Step 1: Start

Step 2: declare variable flag=0

Step 3: open student.dat file in read mode

Step 4: check condition(fread(&data, sizeof(data), 1, file))>0),if true then go to step 5 otherwise go to step 9

Step 5: check condition(data.rollno == n),if true then go to step 6 otherwise go

to step 8 Step 6: call display function

Step 7: flag = 1

Step 8: go to step 4

Step 9: check condition(flag == 0),if true then go to step 10 otherwise go to

step 11 Step 10: print record does not exist

Step 11: Stop
void modify_student()

Step 1: Start

Step 2: Declare variables

no,found=0 Step 3: read value for

roll number

Step 4: open file student.dat in read mode

Step 5: check condition((fread(&data, sizeof(data), 1, file)) > 0 && found == 0),if true then go to step 6 otherwise go to step 13

Step 6: check condition(data.rollno == no),if true then go to step 7 otherwise go to step 12 Step 7: call display function

Step 8: read new details of student

Step 9: set the file pointer to corresponding

location

Step 10: write data to file

Step 11:

found=1 Step

12: go to step

Step 13: check condition(found==0),if true then go to step 14 otherwise go to step

step 15 Step 14: print record not found

Step 15: Stop

void delete_student()

Step 1: Start

Step 2: Declare variables no

Step 3: read roll number to

delete Step 4: open file

student to read

Step 5: open a temporary file to write

Step 6: check condition((fread(&data, sizeof(data), 1, file)) > 0),if true then go to step
7 otherwise go to step 10
Step 7: check condition(data.rollno!=no),if true then go to step 8 otherwise go

to step Step 8: write data to temporary file

Step 9: go to step 6

Step 10: delete file student

Step 11: rename temporary file as

student Step 12:Stop

void class_result()

Step 1: Start

Step 2: open file student to read

Step 3: Check condition(file==NULL),if true then go to step 4 otherwise go to

step 5 Step 4: print file cannot be opened

```
Step 5: Check condition((fread(&data, sizeof(data), 1, file)) > 0),if true then go to step
6 otherwise go to step 8
Step 6: print details of
student Step 7: go to step 5
Step 8: Stop
void result_menu()
Step 1: Start
Step 2: Declare variables rno,ans,choice
Step 3: Read value for choice and check
cases Step 4: case 1: call function result
case 2: read roll number of student and print
report Step 5: Stop
void entry_menu()
Step 1: Start
Step 2: print create,display,search,modify,delete student
records Step 3: read value for choice and check cases
Step 4: case 1: call write_student
function case 2: call display_all
function
case 3: declare variable num
read value for num and call display_sp
function case 4: call modify_student
function
case 5: call delete_student
function default: call
entry_menu function Step 5:
Stop
```

Program

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

```
student
{
   int rollno;
   char name[50];
   int test_score1, test_score2,
   test_score3; float average;
   char grade;
};
   FILE *file;
char calculate_grade(float
average); void input(struct
student *s);
   void display(struct student
   st); void write_student();
   void
   display_all();
   void
   display_sp(int n);
   void
   modify_student()
   ; void
   delete_student();
   void
   class_result();
   void
   result_menu();
   void
   entry_menu();
   int main()
{
   int
   choic
   e;
   clrscr
   (); do
{
   printf("\n\nMain Menu");
   printf("\n1. Result Menu");
   printf("\n2. Entry/Edit
   Menu"); printf("\n3. Exit");
printf("\nPlease select your choice (1-3):
"); scanf("%d",&choice);
   switch (choice)
{
   case 1:
   result_me
   nu();
   break;
   case 2:
   entry_menu();
}
   }while (choice
   != 3); return 0;
}
   char calculate_grade(float average)
{
   if (average >= 90)
   return 'A';
   else if (average >=
   70) return 'B';
   else if (average >= 50)
```

```
return 'C';
else
return
'F';
}
  void input(struct student *s)
{
  printf("\nEnter the roll number of
  student: "); d", &s->rollno);
  fflush(stdin);
  printf("\nEnter the Name of
  student: "); gets(s->name);
  printf("\nEnter the marks in test
  1: "); scanf("%d", &s-
  >test_score1); printf("\nEnter the
  marks in test 2: "); scanf("%d",
  &s->test_score2); printf("\nEnter
  the marks in test 3: ");
  scanf("%d", &s->test_score3);
  s->average = (s->test_score1 + s->test_score2 + s-
  >test_score3) / 3.0; s->grade = calculate_grade(s->average);
}
  void display(struct student st)
{
  printf("\nRoll Number of student : %d",
  st.rollno); printf("\nName of student : %s",
  st.name); printf("\nScore in test 1 : %d",
  st.test_score1); printf("\nScore in test 2 : %d",
  st.test_score2);

  printf("\nScore in test 3 : %d", st.test_score3);
  printf("\nAverage score : %0.2f", st.average);
  printf("\nGrade : %c", st.grade);
}
  void write_student()
{
  struct student data;
file = fopen("student.dat", "a");

  printf("\n\nPlease enter the details of
  student \n"); input(&data);
  fwrite(&data, sizeof(data), 1,
  file); fclose(file);
  printf("\nStudent Record Has Been Created ");
}
  void display_all()
{
  struct student data;
  printf("\n\nDISPLAY ALL RECORD
  !!!\n"); file = fopen("student.dat",
  "rb");
  while ((fread(&data, sizeof(data), 1, file)) > 0)
{


  display(data);
  printf("\n====================================
  \n");
  } fclose(file);
}
  void display_sp(int n)
{
  struct student
  data; int flag
  = 0;
  file = fopen("student.dat", "rb");
  while ((fread(&data, sizeof(data), 1, file)) > 0)
{
  if (data.rollno == n)
{
  display(da
  ta); flag =
  1;
}}
  fclose(fi
  le);

  if (flag
  == 0)
  printf("\nRecord not exist");
}
  void modify_student()
{
  struct student
  data; int no,
  found = 0;
  printf("\nTo Modify ");

  printf("\nPlease Enter The roll number of student:
  "); scanf("%d", &no);
  file = fopen("student.dat", "rb+");
  while ((fread(&data, sizeof(data), 1, file)) > 0 && found == 0)
{
  if (data.rollno == no)
{
  display(data);
  printf("\nPlease enter the new details of student
  \n"); input(&data);
  fseek(file, -
  (long)sizeof(data), 1);
  fwrite(&data, sizeof(data),
  1, file); printf("\n Record
  Updated"); found = 1;
}}
  fclose(file);
  if (found == 0)
  printf("\n Record Not Found ");
}
  void delete_student()
{
int no;
  struct student
  data; FILE
  *file2;
  printf("\n\nDelete Record");
  printf("\nPlease Enter The roll number you want to
  delete: "); scanf("%d", &no);
  file = fopen("student.dat",
  "rb"); file2 =
  fopen("temp.dat", "wb");
  rewind(file);
  while ((fread(&data, sizeof(data), 1, file)) > 0)
{
  if (data.rollno != no)
{
  fwrite(&data, sizeof(data), 1, file2);
}
}
  fclose(file2);
  fclose(file);
  remove("student.d
  at");
  rename("temp.dat",
  "student.dat");
  printf("\nRecord deleted.");
}
  void class_result()
{
  struct studentdata;
  file = fopen("student.dat",
  "rb"); if (file == NULL)
{
  printf("ERROR!!! FILE COULD NOT BE OPEN\n\n Go To Entry Menu to
  create File"); printf("\n\n Program is closing          ");
  return;
}
  printf("\nALL STUDENTS RESULT \n");
  printf("====================================================\n");
  printf("R.No.\tName\t\tTest1\tTest2\tTest3\tAverage\tGrade\n");
  printf("========================================================\n");
  while ((fread(&data, sizeof(data), 1, file)) > 0)
{
  printf("%-7d %-15s %-7d %-7d %-7d %-7.2f %-1c\n",
  data.rollno, data.name, data.test_score1, data.test_score2,
  data.test_score3, data.average, data.grade);
```

```
}
   fclose(file);
}
   void result_menu()
{
   int rno,
   ans;
   char
   choice;
   printf("\n\nResult Menu");
   printf("\n1. Class Result\n2. Student Report Card\n3.Back to Main
   Menu"); printf("\nEnter Choice (1-3)? ");
   scanf("%d",
   &ans); switch
   (ans)
{
   case 1:
   class_resu
   lt();
   break;
   case 2:
   do
   {ans;
printf("\n\nEnter roll number of student:
"); scanf("%d", &rno);
   display_sp(rno);
   printf("\nDo you want to see more result (y/n)?: ");
   fflush(stdin);
   scanf("%c", &choice);
   }while (choice == 'y' || choice ==
   'Y'); break;
   case 3:
   break;
   default:
   printf("\
   a");
 }
 }
   void entry_menu()

 {
   int choice;
   printf("\n\nEntry
   Menu");
printf("\n1.Create Student Record");
printf("\n2.Display all students
records"); printf("\n3.Search student
record "); printf("\n4.Modify student
record"); printf("\n5.Delete Student
record");
   printf("\n6.Back to main menu");
   printf("\nEnter your choice (1-6): ");
   scanf("%d",&choice);
   switch (choice)
 {
   case 1:
   write_studen
   t(); break;
   case 2:
   display_al
   l();
```

```
  break;
  case 3:
{
  int num;
  printf("\n\nPlease enter the roll
  number: "); scanf("%d", &num);
  display_sp(num);
} break; case 4:
  modify_student(
  ); break;
  case 5:
  delete_student();
  break;
case 6:
break;
  default:
  printf("\a");
  entry_menu();
}
}
```

Output

```
   Main Menu
1. Result Menu
2. Entry/Edit Menu
3. Exit
   Please select your choice (1-
   3): 2 Entry Menu
1. Create Student Record
2. Display all students records
3. Search student record
4. Modify student record
```

```
5. Delete Student record
   6.  Back to main
   menu Enter your
   choice (1-6): 1
   Please enter the details of
   student Enter the roll number of
   student: 103 Enter the Name of
   student: Kabir Enter the marks
   in test 1: 95
   Enter  the marks  in  test  2: 85
   Enter the marks in test 3: 91
   Student Record Has Been
   Created Main Menu

1. Result Menu
2. Entry/Edit Menu
3. Exit
   Please select your choice (1-
   3): 2 Entry Menu
1. Create Student Record
2. Display all students records
3. Search student record

4. Modify student record
5. Delete Student record
   6.  Back to main
   menu Enter your
   choice (1-6): 3
   Please enter the roll number:
   103 Roll Number of student
   : 103 Name of student :
```

```
   Kabir

   Score in test 1 : 95

   Score in test 2 : 85

   Score in test 3 : 91

   Average score : 90.33

   Grade : A

   Main Menu

1. Result Menu

2. Entry/Edit Menu

3. Exit

   Please select your choice (1-

   3): 1 Result Menu

 1.  Class Result

    2. Student Report Card

 2.  3.Back to Main Menu
 3.  Enter Choice (1-3)? 1
 4.  ALL STUDENTS RESULT
 5.      ==========================================================

 6.  R.No. Name Test1 Test2 Test3 Average Grade
 7.      ==========================================================
 8.  100 Alex 45 78 65 62.67 C
 9.  101 Jai 78 95 92 88.33 B
 10. 103 Kabir 95 85 91 90.33 A
 11. Main Menu
 12. 1. Result Menu
 13. 2. Entry/Edit Menu
 14. 3.Exit
 15. Please select your choice (1-3): 3
```

17. Write the following functions that,

i. Request the user for two integer numbers and output them and their sum.

ii. Request the user for two integers and output their remainder after division.

iii. **Request the user for two floats and output their product.**

iv. **Request the user for a word and print it twice on the same row.**

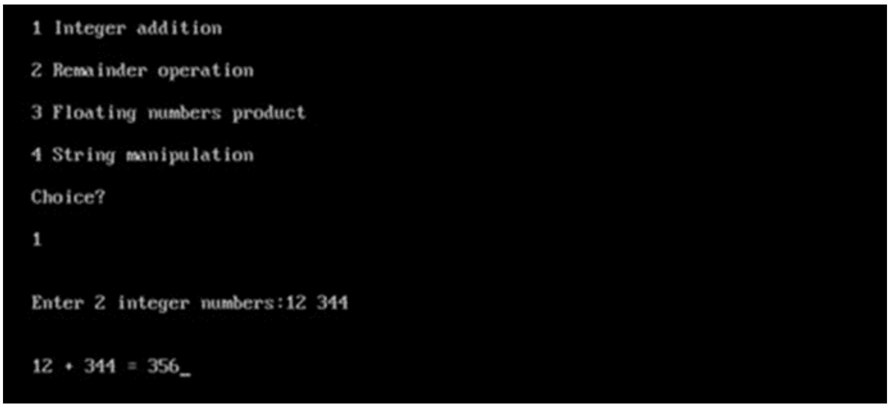
Write a C program to provide the above functions as options to the user using switch statement and perform the functions accordingly.

Program

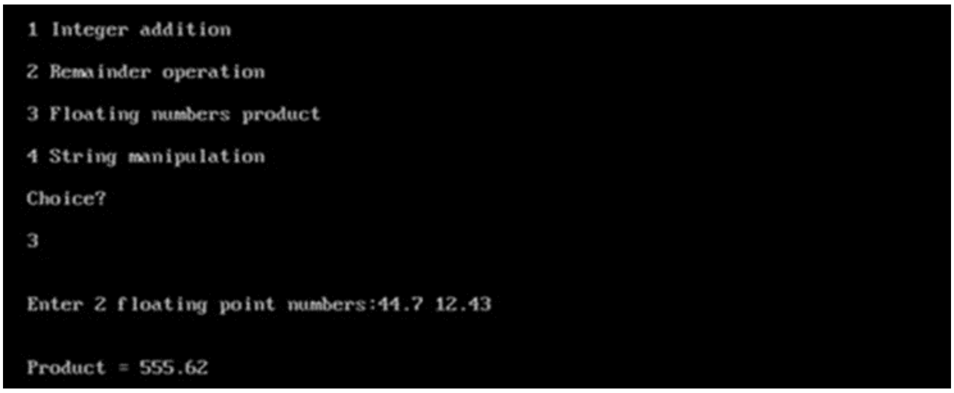
```
#include<stdio.h>
#include<conio.h>
  void sum()
{
 int a,b;
  printf("\n\nEnter 2 integer
  numbers:"); scanf("%d
  %d",&a,&b); printf("\n\n%d +
  %d = %d",a,b,a+b);
}
  void remainder()
int a,b;
  printf("\n\nEnter 2 integer numbers:");
  scanf("%d %d",&a,&b);
  printf("\n\nRemainder = %d",a%b);
}
  void float_mul()
{
 float a,b;
  printf("\n\nEnter 2 floating point
  numbers:"); scanf("%f %f",&a,&b);
  printf("\n\nProduct = %.2f",a*b);
}
```

```
   void str_manipulation()

{
 char str[20];
printf("\n\nEnter a string
:"); scanf("%s",str);
printf("\n\n%s
%s",str,str);
}
   void main()
{
    int ch;
   clrscr();
   printf("\n\n1 Integer addition\n\n2 Remainder operation\n\n3 Floating numbers
   product\n\n4 String manipulation\n\nChoice?\n\n");
scanf("%d",&ch);
 switch(ch)
 { case 1: sum();break; case 2:
remainde
r();
break;
case 3:
   float_m
   ul();
   break;
     case 4:
     str_manipulatio
   n(); break;
  default:
printf("\n\nInvalid
  entry!!"); break;
 }
  getch();
}
```

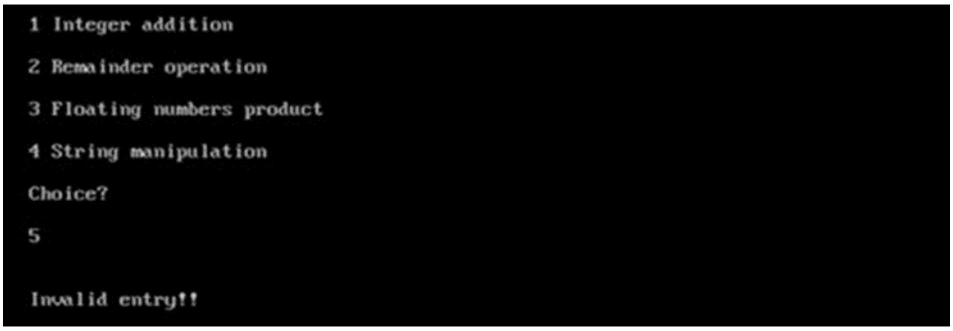
Output



```
1 Integer addition
2 Remainder operation
3 Floating numbers product
4 String manipulation
Choice?
1

Enter 2 integer numbers:12 344

12 + 344 = 356_
```



```
1 Integer addition
2 Remainder operation
3 Floating numbers product
4 String manipulation
Choice?
3

Enter 2 floating point numbers:44.7 12.43

Product = 555.62
```



```
1 Integer addition
2 Remainder operation
3 Floating numbers product
4 String manipulation
Choice?
5

Invalid entry!!
```

18. write a program to generate the sum of series as shown below

program

```
#include<stdio.h
>   int   add(int
num)
{
    int i,j,fact,sum=0;
    for(i=1;i<=num;i++)//loop for finding factorial and sum
    {
        fact=1;


        if(i!=num)
            printf("%d!+ ",i);
        else
            printf("%d!= ",i);
        for(j=1;j<=i;j++)
            fact=fact*j;
        sum=sum+fact;
    }
    return sum;
}
int main()
{
    int num;
    printf("Enter the last number of series:\n");
    scanf("%d",&num);//last number of series
    printf("%d",add(num));
    return 0;
}
```

Output:

```
Enter the last number of series:10
1!+ 2!+ 3!+ 4!+ 5!+ 6!+ 7!+ 8!+ 9!+ 10!= 4037913
```

19. write a program to store all the even numbers and odd numbers starting from 2 to 50 separately in two different arrays namely even and odd .The content of both the arrays should be displayed back to the screen .

program

```
#include<stdio
   h>

 void main()
   {
      long int ARR[10], OAR[10], EAR[10];
      int i, j = 0, k = 0, n;

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

scanf("%d", &n);


printf("Enter the elements of the array
      n"); for (i = 0; i< n; i++)
      {
scanf("%ld", &ARR[i]);
fflush(stdin);
}

      for (i = 0; i< n; i++)
      {
         if (ARR[i] % 2 == 0)
         {
j++;EAR[j] = ARR[i];

}  else
         {
                OAR[k] = ARR[i];

                k++;
            }
        }
printf("The elements of OAR are
        n"); for (i = 0; i< k; i++)
        {
printf("%ldn", OAR[i]);
        }
printf("The elements of EAR are
        n"); for (i = 0; i< j; i++)
        {
printf("%ldn", EAR[i]);
        }
    }
```

Output

```
Enter the size of array
AR 6
Enter the elements of the
array 34
56
78
90
12
39
The elements of OAR
are

39
The elements of EAR
are 34
```

```
56

78

90

12
```

20. Write a program to find the matrix multiplication using functions.

program

```
#include<stdio.
h>
#include<stdlib.
h> int main(){
int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
system("cls");
printf("enter the number of row=");
scanf("%d",&r);
printf("enter the number of column=");
scanf("%d",&c);
printf("enter the first matrix
element=\n"); for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("enter the second matrix
element=\n"); for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&b[i][j]);

}
}


printf("product of the
matrix=\n"); for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
mul[i][j]=0;
for(k=0;k<c;k++)
{
mul[i][j]+=a[i][k]*b[k][j];
}
}
}

//for printing
result
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
printf("%d\t",mul[i][j]);
}
printf("\n");
}
return 0;
```

}

Output:

```
enter the number of
row=3 enter the number of
column=3
enter the first matrix
element=
  1 1 1
  2 2 2
  3 3 3

  enter the second matrix element=
  1 1 1
  2 2 2
  3 3 3
  product of the
  matrix= 6 6 6
  12 12 12
  18 18 18
```

MINI PROJECT

TITLE: Building A Calendar Using C

1. INTRODUCTION

This project is a simple calendar program built using the C programming language. The program takes a year as input from the user and displays the calendar for that year. It accounts for leap years, ensuring that February has 29 days when necessary. The calendar is displayed with days of the week (Sunday to Saturday) aligned correctly under each month. This project is designed to help users visualize the calendar for any given year.

The core concepts utilized in this project are:

- Handling user input
- Calculating the starting day of the year
- Displaying formatted output
- Working with arrays for months and days
- Leap year calculation

2. OBJECTIVE

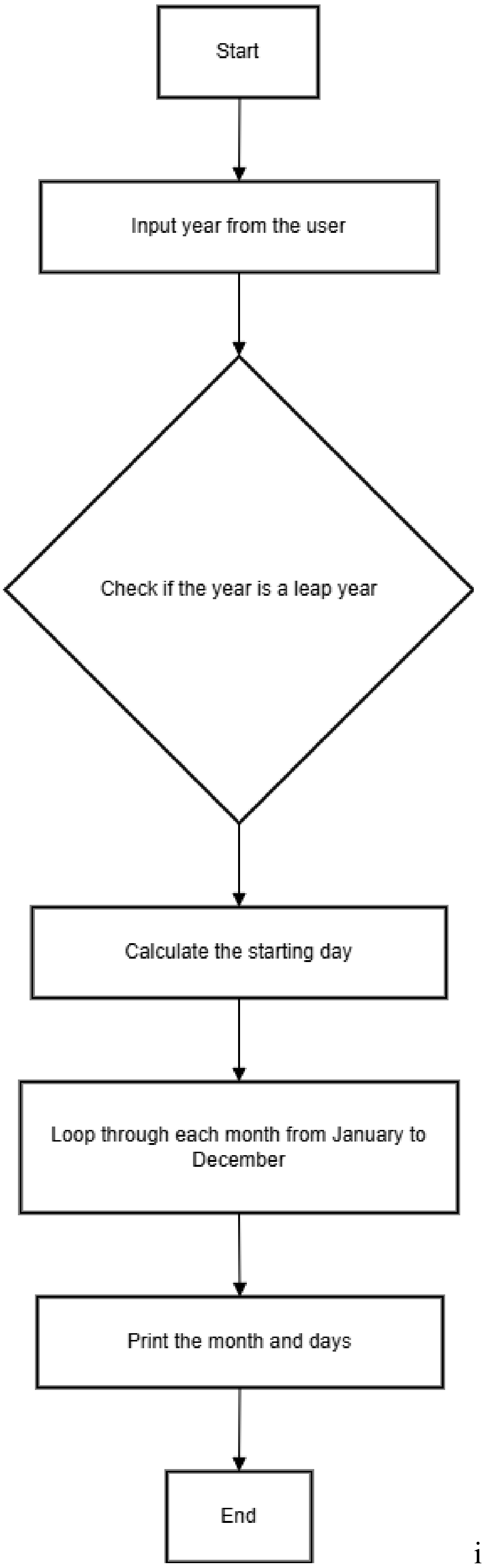
The main objective of this project is to implement a simple and efficient calendar generator in C. The program allows the user to:

1. Input a specific year.
2. View the calendar for each month in that year.
3. Automatically adjust for leap years (February having 29 days if the year is a leap year).
4. Display the days of the week under each month in the correct alignment.

By completing this project, the goal is to practice handling dates, arrays, and basic date calculations in C.

3. FLOWCHART

Here’s a simplified flowchart to represent the logic of your calendar program:


Start
Input year from the user
Check if the year is a leap year
Calculate the starting day
Loop through each month from January to December
Print the month and days
End

ies

4. MODULES DESCRIPTION

The project "Calendar Generator" has two main modules:

4.1 Get First Weekday (get_1st_weekday)

This module calculates the weekday of the first day of the given year. It uses a formula based on the total

number of days that have passed since a base year (including adjustments for leap years). The module

returns an integer representing the starting weekday for January 1st of the specified year.

Key Functions:

- Input: Year (integer)
- Output: Weekday of the first day of the year (0 for Sunday, 1 for Monday, etc.)
- Description: This function calculates the weekday of January 1st by applying a mathematical formula considering leap years and the Gregorian calendar system**.**

4.2 Main Calendar Generator (main)

This module is responsible for generating and displaying the calendar for the specified year. It takes the user's input year, calculates the starting weekday, adjusts for leap years, and then prints the calendar for each month in the year. The calendar displays the days of the week (Sunday to Saturday) for each month and aligns the days under the correct weekday header.

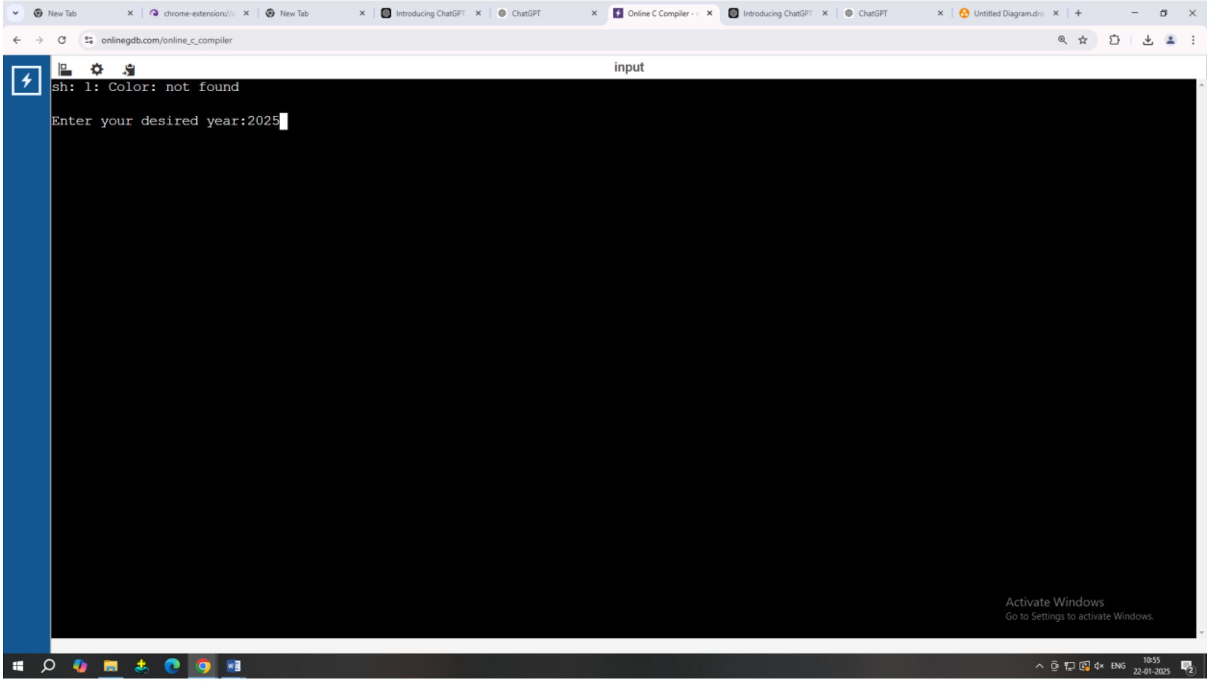
Key Functions:

- **Input**: Year (integer)
- **Output:** A formatted calendar for the specified year (displayed in the terminal).
- **Description**: This module handles the user interaction, including:
    - Prompting the user for the desired year.
    - Checking if the year is a leap year and adjusting February's days if necessary.
    - Looping through each month, displaying the month name and day layout, formatted under the correct weekdays.

- Updating and printing the calendar month by month.

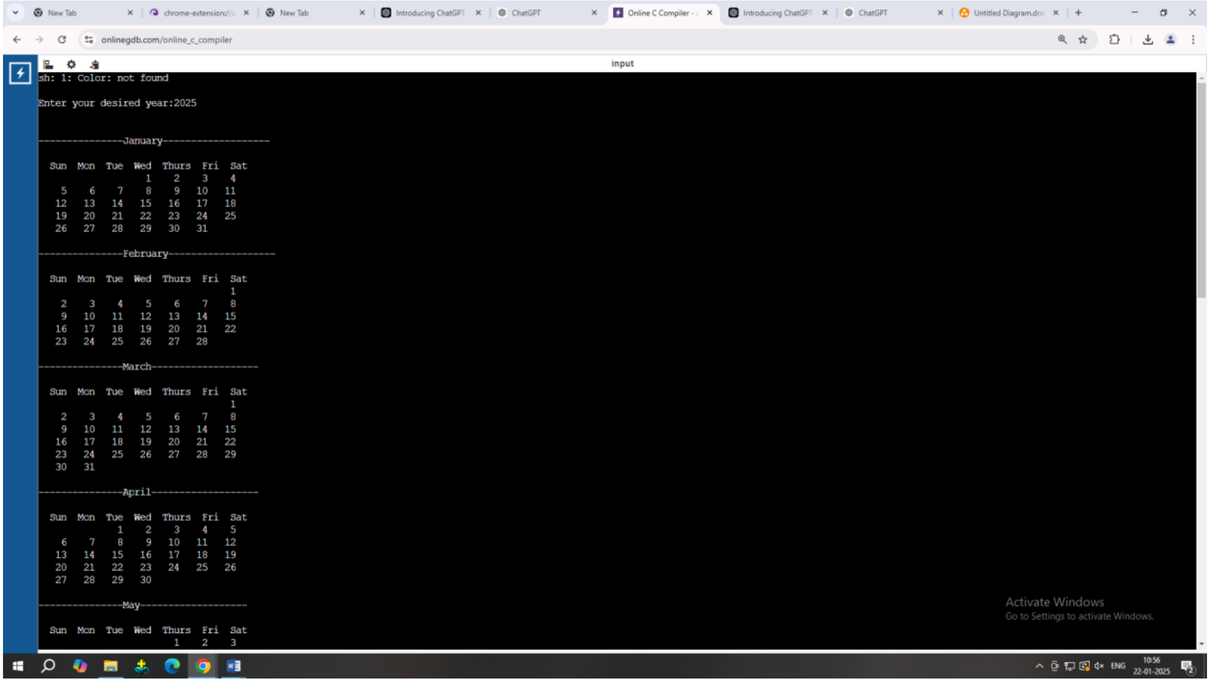
5. INPUT AND OUTPUT

5.1. Input

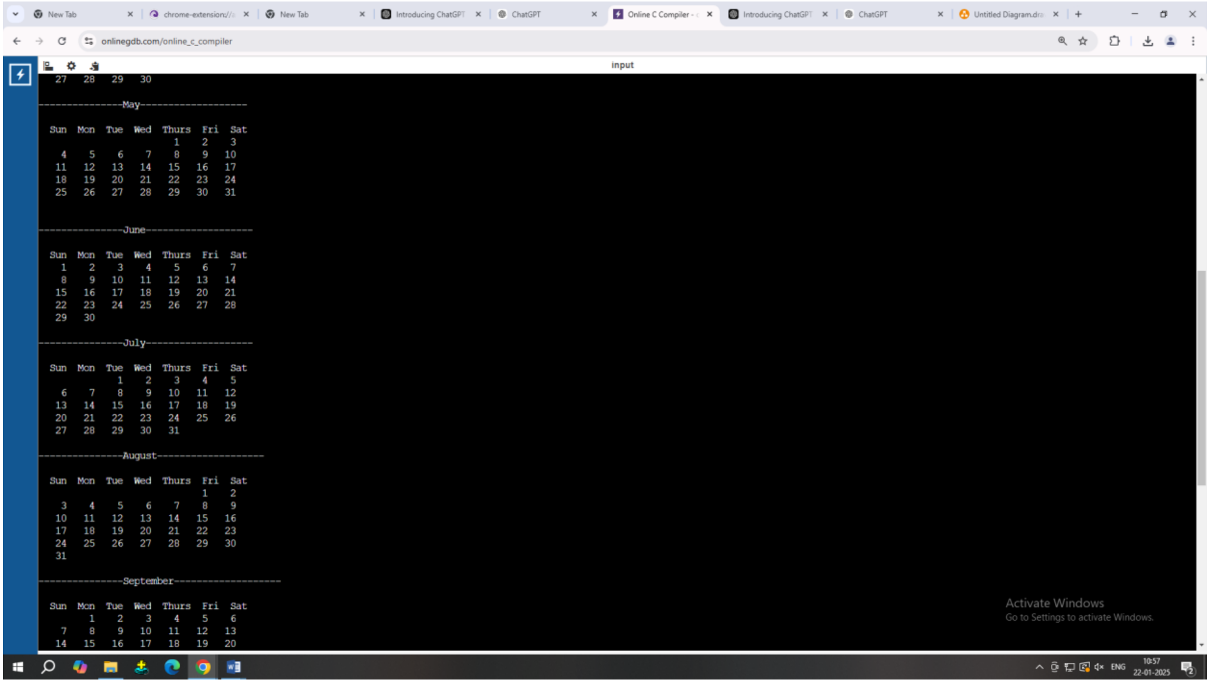

onlinegdb.com/online_c_compiler
input
Activate Windows
Go to Settings to activate Windows.

```
sh: 1: Color: not found

Enter your desired year:2025
```

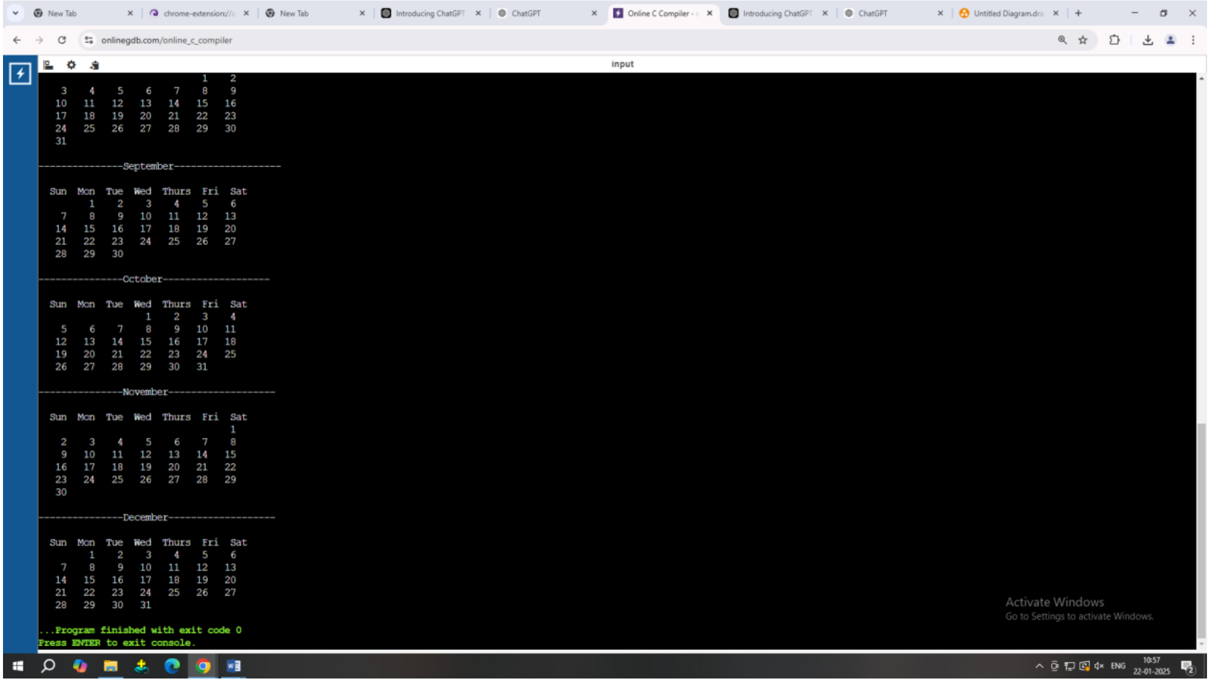
5.2. Outputs


onlinegdb.com/online_c_compiler
input
Activate Windows
Go to Settings to activate Windows.

```
sh: 1: Color: not found

Enter your desired year:2025


---------------January--------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
                   1      2    3    4
    5    6    7    8      9   10   11
   12   13   14   15     16   17   18
   19   20   21   22     23   24   25
   26   27   28   29     30   31

---------------February--------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
                                    1
    2    3    4    5      6    7    8
    9   10   11   12     13   14   15
   16   17   18   19     20   21   22
   23   24   25   26     27   28

---------------March-------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
                                    1
    2    3    4    5      6    7    8
    9   10   11   12     13   14   15
   16   17   18   19     20   21   22
   23   24   25   26     27   28   29
   30   31

---------------April-------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
              1    2      3    4    5
    6    7    8    9     10   11   12
   13   14   15   16     17   18   19
   20   21   22   23     24   25   26
   27   28   29   30

---------------May-----------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
                          1    2    3
```



```
   27   28   29   30

--------------May------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
                        1    2    3
    4    5    6    7    8    9   10
   11   12   13   14   15   16   17
   18   19   20   21   22   23   24
   25   26   27   28   29   30   31


--------------June------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
    1    2    3    4    5    6    7
    8    9   10   11   12   13   14
   15   16   17   18   19   20   21
   22   23   24   25   26   27   28
   29   30

--------------July------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
              1    2    3    4    5
    6    7    8    9   10   11   12
   13   14   15   16   17   18   19
   20   21   22   23   24   25   26
   27   28   29   30   31

--------------August------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
                             1    2
    3    4    5    6    7    8    9
   10   11   12   13   14   15   16
   17   18   19   20   21   22   23
   24   25   26   27   28   29   30
   31

--------------September------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
         1    2    3    4    5    6
    7    8    9   10   11   12   13
   14   15   16   17   18   19   20
```



```
                             1    2
    3    4    5    6    7    8    9
   10   11   12   13   14   15   16
   17   18   19   20   21   22   23
   24   25   26   27   28   29   30
   31

--------------September------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
         1    2    3    4    5    6
    7    8    9   10   11   12   13
   14   15   16   17   18   19   20
   21   22   23   24   25   26   27
   28   29   30

--------------October------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
                   1    2    3    4
    5    6    7    8    9   10   11
   12   13   14   15   16   17   18
   19   20   21   22   23   24   25
   26   27   28   29   30   31

--------------November------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
                                  1
    2    3    4    5    6    7    8
    9   10   11   12   13   14   15
   16   17   18   19   20   21   22
   23   24   25   26   27   28   29
   30

--------------December------------------

  Sun  Mon  Tue  Wed  Thurs  Fri  Sat
         1    2    3    4    5    6
    7    8    9   10   11   12   13
   14   15   16   17   18   19   20
   21   22   23   24   25   26   27
   28   29   30   31

...Program finished with exit code 0
Press ENTER to exit console.
```

6. CONCLUSION

This calendar program in C provides an easy way for users to view a calendar for any year, accounting for leap years. Through this project, I gained a deeper understanding of working with arrays, conditional logic, and basic date calculations in C. The program successfully implements a well-formatted calendar, displaying each month with the correct number of days and aligned weekdays.

Challenges faced:

- Ensuring leap year rules were correctly applied.
- Formatting the output to display days of the week correctly aligned.

Future improvements:

- Extend the program to allow the user to display calendars for multiple years.
- Add the option to display specific dates or holidays.
- Enhance the user interface with graphical representation.

7. REFERENCES

- **C Programming Language - The Complete Reference** by Herbert Schildt
  This book provided essential information on C programming, syntax, and functions.
- **GeeksforGeeks**
  Article on leap year calculations: https://www.geeksforgeeks.org
- **W3Schools - C Programming**
  A helpful guide on basic C programming concepts: https://www.w3schools.com/c/
- **Stack Overflow**
  A community platform used for troubleshooting specific coding issues: https://stackoverflow.com/
- **TimeandDate.com**
  A website used for understanding calendar formatting: https://www.timeanddate.com/calendar/

8. APPENDIX

SOURCE CODE

```
#include <stdio.h>
#include <stdlib.h>
int get_1st_weekday(int year)
{
  int d;
  d = (((year - 1) * 365) + ((year - 1) / 4) - ((year - 1) / 100) + ((year) / 400) + 1) % 7;
  return d;
}
int main()
{
   system("Color 3F");
   int year,month,day,daysInMonth,weekDay=0,startingDay;
   printf("\nEnter your desired year:");
   scanf("%d",&year);
   char *months[]={"January","February","March","April","May","June","July","August","September","October","November","December"};
   int monthDay[]={31,28,31,30,31,30,31,31,30,31,30,31};
   if((year%4==0&&year%100!=0)||year%400==0)
   {
       monthDay[1]=29;
   }
   startingDay=get_1st_weekday(year);
   for(month=0;month<12;month++)
   {
      daysInMonth=monthDay[month];
      printf("\n\n---------------%s-------------------\n",months[month]);
   printf("\n  Sun  Mon  Tue  Wed  Thurs  Fri  Sat\n");
   for(weekDay=0;weekDay<startingDay;weekDay++)
   {
    printf("     ");
   }
   for(day=1;day<=daysInMonth;day++)
   {
    printf("%5d",day);
    if(++weekDay>6)
     {
       printf("\n");
       weekDay=0;
     }
    startingDay=weekDay;
   }
  }
}
```