

Experiment no – 01(a)

Aim: Write an algorithm and draw flowchart for Area of circle.

Algorithm:

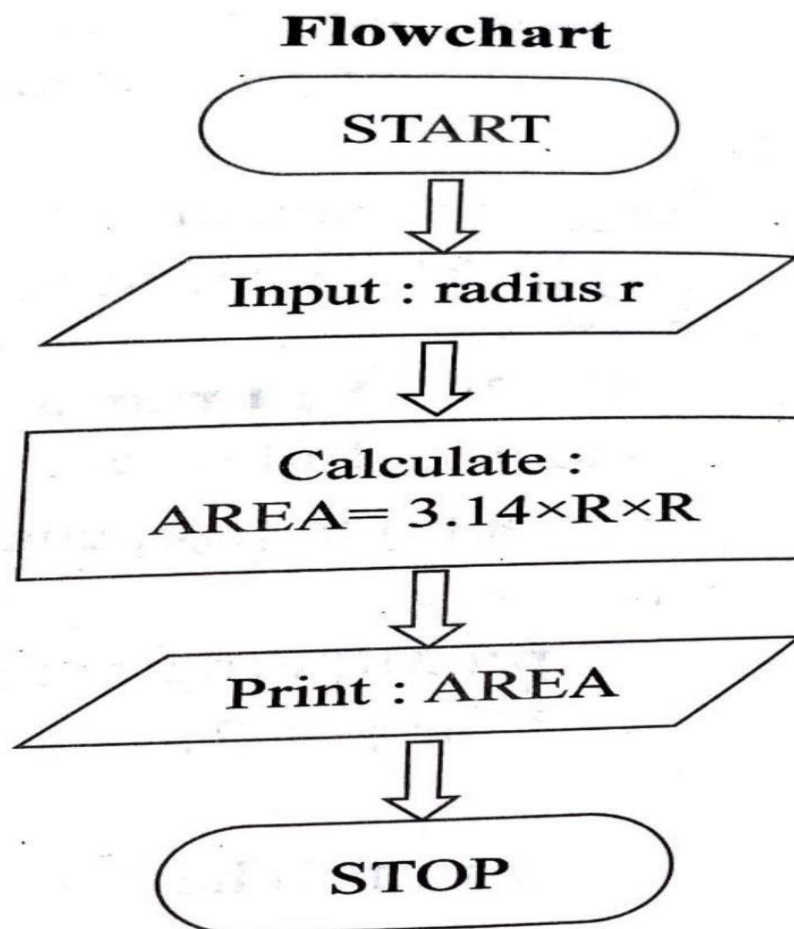
Step 1: Start

Step 2: Read the circle's radius R value.

Step 3: Calculate area of circle i.e. $AREA = 3.14 \times R \times R$

Step 4: Print AREA

Step 5: Stop **Flowchart:**



Conclusion : Successfully Drawn flowchart and wrote an algorithm

Experiment no – 01(b)

Aim: Write an algorithm and draw flowchart to print the given no. is even or odd.

Algorithm:

Step 1: Start

Step 2: Read the number value NUM.

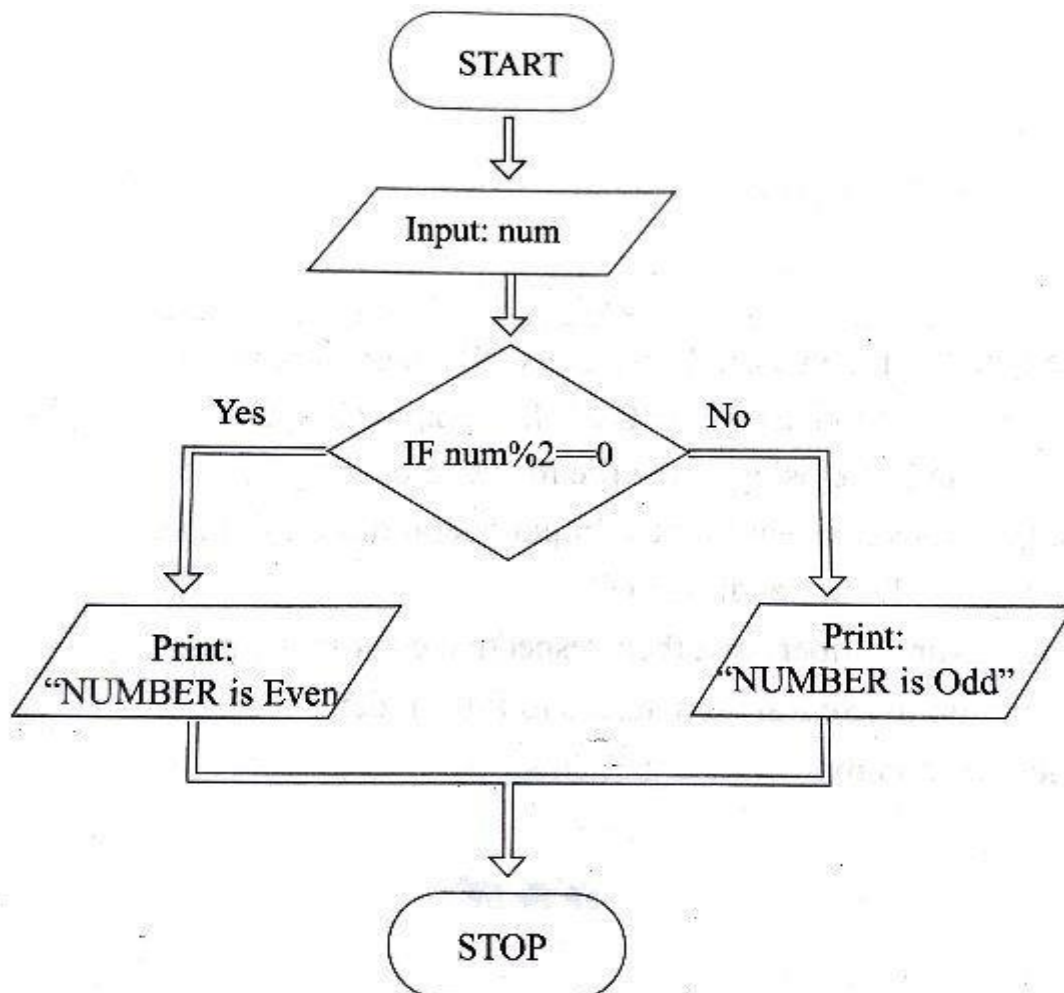
Step 3: Divide the NUM by 2 and store the remainder in REM

Step 4: If REM = 0 Then go to Step 6

Step 5: Print "NUMBER is Odd" go to step 7

Step 6: Print "NUMBER is Even"

Step 7: Stop **Flowchart:**



Conclusion : Successfully Drawn flowchart and wrote an algorithm.

Experiment no – 01(c)

Aim: Write an algorithm and draw flowchart to print 1 to 10 numbers.

Algorithm:

Step 1: Start

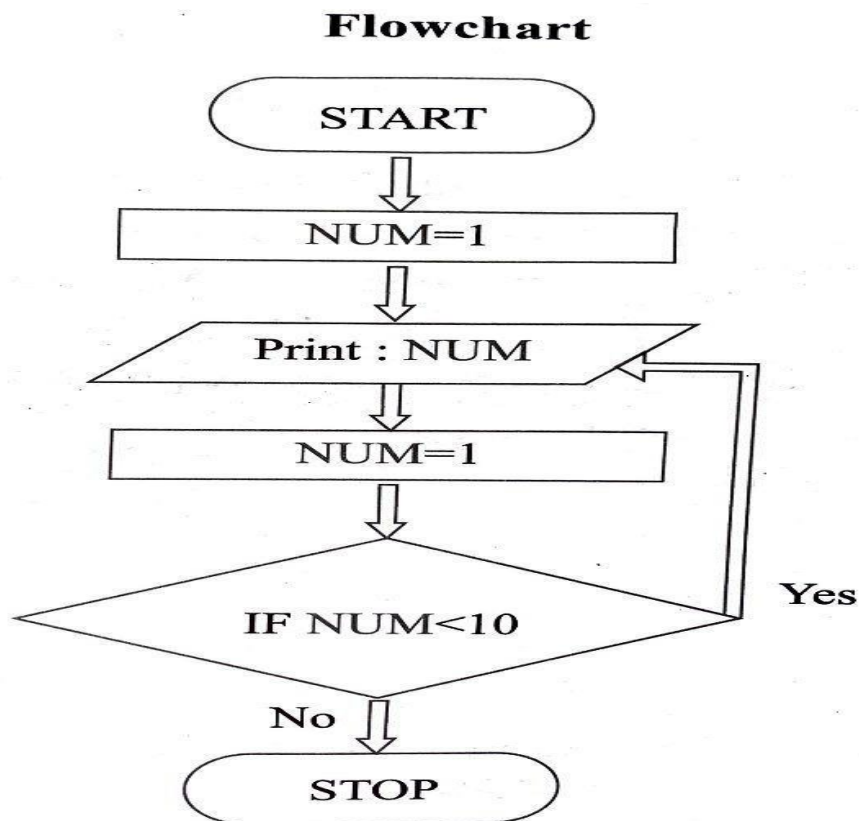
Step 2: Initialize the variable NUM = 1

Step 3: Print NUM

Step 4: Increment NUM by 1 NUM=NUM+1

Step 5: If NUM <= 10 go to Step 3

Step 6: Stop **Flowchart:**



Conclusion : Successfully Drawn flowchart and wrote an algorithm.

Experiment no – 01(d)

Aim: Write an algorithm and draw flowchart for sum of 1 to 5 numbers.

Algorithm:

Step 1: Start

Step 2: Initialize the variable NUM = 1 and SUM=0

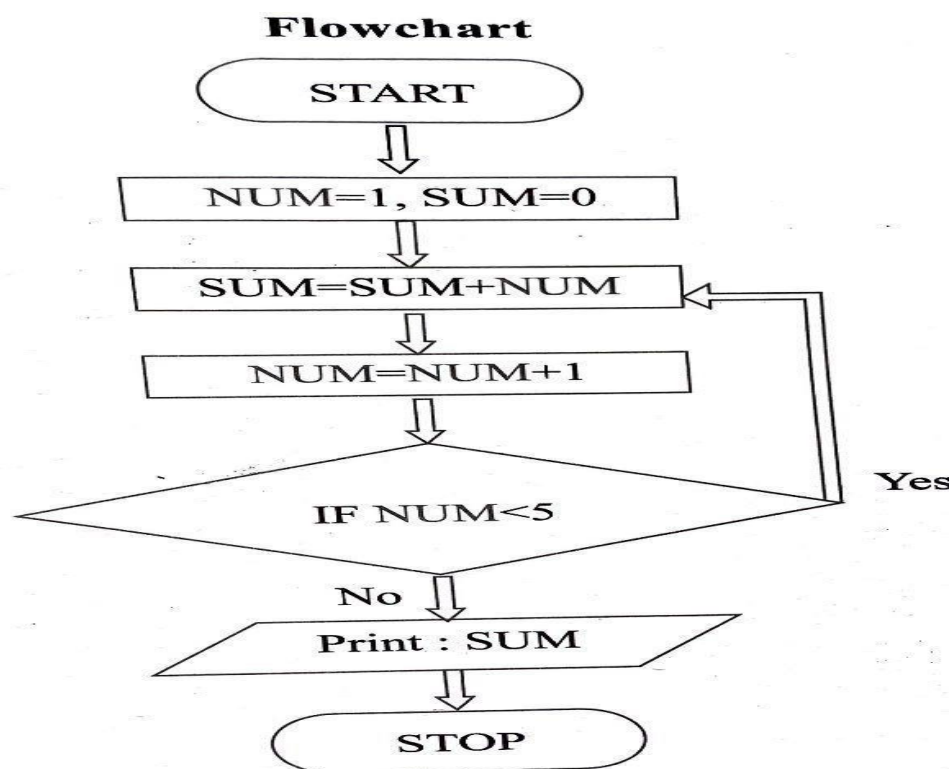
Step 3: SUM=SUM+NUM

Step 4: Increment NUM by 1 NUM=NUM+1

Step 5: If NUM<=5 go to Step 3

Step 6: Print SUM

Step 7: Stop **Flowchart:**



Conclusion : Successfully Drawn flowchart and wrote an algorithm.

Experiment no – 01(e)

Aim: Write an algorithm and draw flow chart to compute the addition of digits of a given number.

Algorithm:

Step 1: Start

Step 2: Read the number value NUM

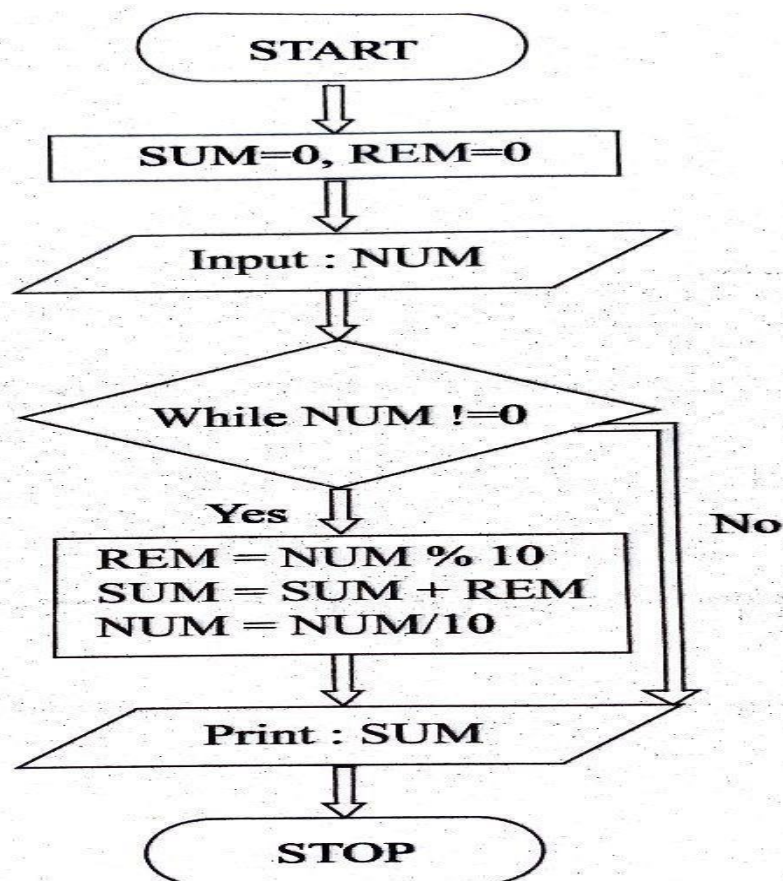
Step 3: Initialize SUM = 0

Step 4: Perform $REM = NUM \% 10$ and add REM to SUM i.e. $SUM = SUM + REM$

Step 5: Perform $NUM = NUM/10$

Step 6: IF $NUM = 0$ stop the process and Print SUM else go to Step 3

Step 7: Stop **Flowchart:**



Conclusion : Successfully Drawn flowchart and wrote an algorithm.

Experiment no – 02(a)

Aim: Write a program using while loop to reverse the digits of a number.

Algorithm:

- i. Ask the user to enter any number.
- ii. Declare and initialize another variable reversed with 0, where reversed an integer variable.
- iii. Get the last digit of the given number by performing the modulo division (%) and store the value in last_digit variable, likey last_digit= number % 10.
- iv. Multiply reversed by 10 and add last_digit, like reversed = reversed*10 + last_digit.
- v. Divide numbered by 10, like numbered/10.
- vi. Repeat the steps 3 to 5 till numbered is not equal to (or greater than) zero.

Code:

```
#include <stdio.h>

int main() {
printf("13_Khushi Tiwari\n");
int num, rnum = 0, rem;
printf("Enter any number: ");
scanf("%d", &num);
while (num != 0) {
    rem = num % 10;
    rnum = rnum * 10 + rem;
    num = num / 10; }
printf("\nReverse of input number is: %d", rnum);
return 0;}
```

Output:

```
/tmp/pzyoto5Uh3.o
13_Khushi Tiwari
Enter any number: 13
Reverse of input number is: 31
```

Conclusion : Successfully performed program using while loop to reverse the digits of a number.

Experiment no – 02(b)

Aim: Write a program to calculate the factorial of a given number.

Algorithm:

- i. Start program
- ii. Ask the user to enter an integer to find the factorial
- iii. Read the integer and assign it to a variable
- iv. From the value of the integer up to 1, multiply each digit and update the final value
- v. The final value at the end of all the multiplication till 1 is the factorial
- vi. End program

Code:

```
#include <stdio.h>

int main() {
    {
        printf("13_Khushi Tiwari\n");}
    int n, i;
    unsigned long long fact = 1;
    printf("Enter an integer: ");
    scanf("%d", &n);
    // shows error if the user enters a negative integer
    if (n < 0)
        printf("Error! Factorial of a negative number doesn't exist.");
    else {
        for (i = 1; i <= n; ++i) {
            fact *= i;
        }
        printf("Factorial of %d = %llu", n, fact);
    }
}
```

```
    return 0;  
}
```

Output:

```
/tmp/ZgWmARzfIg.o  
13_Khushi Tiwari  
Enter a positive integer :3  
Factorial of 3 is 6
```

Conclusion : Successfully performed a program to calculate the factorial of a given number.

Experiment no – 02(c)**Aim: Write a program to find the roots of quadratic equation.****Algorithm:**

- i. Start
- ii. Read a, b, c values
- iii. Compute $d = b^2 - 4ac$
- iv. if $d > 0$ then
 - i. $r1 = \frac{-b + \sqrt{d}}{2a}$
 - ii. $r2 = \frac{-b - \sqrt{d}}{2a}$
- v. Otherwise if $d = 0$ then
 - i. compute $r1 = -b/2a$, $r2 = -b/2a$
 - ii. print r1, r2 values
- vi. Otherwise if $d < 0$ then print roots are imaginary
- vii. Stop

Code:

```
#include<stdio.h>
#include<math.h>
int main()
{
    printf("13_Khushi Tiwari\n");
    float a,b,c,x1,x2,determinant,realpart,imaginaryPart;
    printf("Enter coefficients a,b and c:");
    scanf("%f%f%f",&a,&b,&c);
    determinant=b*b - 4*a*c;
    if (determinant>0)
    {
        x1=(-b + sqrt(determinant))/(2*a);
        x2=(-b - sqrt(determinant))/(2*a);
        printf("Roots are real and different.");
        printf("\n x1=%.3f",x1);
        printf("\n x2=%.3f",x2);
    }
    else if (determinant==0)
```

```
{  
printf("Roots are real and same.");  
x1=(-b+sqrt(determinant))/(2*a);  
printf("\n x1=%.ef",x1);  
printf("\nx2=%.3f",x2);  
}  
Else  
{  
realpart=-b/(2*a);  
imaginaryPart=sqrt(determinant)/(2*a);  
printf("\n Roots are complex and differtent.");  
printf("\n x1=%.3f+%.fi",realpart,imaginaryPart);  
printf("\nx2 = %.3f-%.3fi",realpart,imaginaryPart);  
}  
return 0;  
}
```

Output:

```
/tmp/pzyoto5Uh3.o  
13_Khushi Tiwari  
Enter coefficients a,b and c:4 5 1  
Roots are real and different.  
x1=-0.250  
x2=-1.000|
```

Conclusion : Successfully performed a program to find the roots of quadratic equation.

Experiment no – 02(d)**Aim: Write a program to print the Fibonacci series.****Algorithm:**

- i. START
- ii. Take integer variable A, B, C
- iii. Set A = 0, B = 0
- iv. DISPLAY A, B
- v. C = A + B
- vi. DISPLAY C
- vii. Set A = B, B = C
- viii. REPEAT from 4 - 6, for n times
- ix. STOP

Code:

```
#include <stdio.h>

int main() {
    printf("13_Khushi Tiwari\n")

    int i, n;

    // initialize first and second terms
    int t1 = 0, t2 = 1;

    // initialize the next term (3rd term)
    int nextTerm = t1 + t2;

    // get no. of terms from user
    printf("Enter the number of terms: ");
    scanf("%d", &n);

    // print the first two terms t1 and t2
    printf("Fibonacci Series: %d, %d, ", t1, t2);

    // print 3rd to nth terms
```

```
for (i = 3; i <= n; ++i) {  
    printf("%d, ", nextTerm);  
    t1 = t2;  
    t2 = nextTerm;  
    nextTerm = t1 + t2;  
}  
return 0;  
}
```

Output:

```
/tmp/n2rwZEVWpQ.o  
13_Khushi Tiwari  
Enter the number of terms: 4  
Fibonacci Series: 0, 1, 1, 2,
```

Conclusion : Successfully performed a program to print the Fibonacci series

Experiment no – 03(a)**Aim: Write a program in C to check entered character vowel or consonant.****Algorithm:**

- i. Start
- ii. Declare character type variable ch
- iii. Read ch from User
- iv. // Checking both lower and upper case vowels.
- v. IF (ch == 'a' || ch == 'A' ||
i. ch == 'e' || ch == 'E' ||
ii. ch == 'i' || ch == 'I' ||
iii. ch == 'o' || ch == 'O' ||
iv. ch == 'u' || ch == 'U')
- vi. Print "Vowel"
- vii. ELSE
- viii. Print "Consonant"
- ix. Stop

Code:

```
#include <stdio.h>

int main() {
    char c;
    printf("13_Khushi Tiwari\n");
    int lowercase_vowel, uppercase_vowel;
    printf("Enter an alphabet: ");
    scanf("%c", &c);

    // evaluates to 1 if variable c is a lowercase vowel
    lowercase_vowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

    // evaluates to 1 if variable c is a uppercase vowel
    uppercase_vowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');

    // evaluates to 1 (true) if c is a vowel
    if (lowercase_vowel || uppercase_vowel)
```

```
    printf("%c is a vowel.", c);  
else  
    printf("%c is a consonant.", c);  
return 0;  
}
```

Output:

```
/tmp/c1fos5GSxZ.o  
13_Khushi Tiwari  
Enter an alphabet: f  
f is a consonant.
```

Conclusion : Successfully performed a program in C to check entered character vowel or consonant.

Experiment no – 03(b)

Aim: Write a program to C program to print day name of week using switch-case.

Algorithm:

- i. Input day number from user. Store it in some variable say *no*.
- ii. Switch the value of *week* i.e. use switch(*no*) and match with cases.
- iii. There can be 7 possible values(choices) of *week* i.e. 1 to 7. Therefore write 7 case inside switch. In addition, add default case as an else block.
- iv. For case 1: print “MONDAY”, for case 2: print “TUESDAY” and so on. Print “SUNDAY” for case 7:.
- v. If any case does not matches then, for default: case print “Invalid week number”.

Code:

```
#include <stdio.h>

int main()
{ printf("13_Khushi Tiwari\n");
  int week;

  /* Input week number from user */
  printf("Enter week number(1-7): ");
  scanf("%d", &week);

  switch(week)
  {
    case 1:
      printf("Monday");
      break;
    case 2:
      printf("Tuesday");
      break;
    case 3:
      printf("Wednesday");
```

```
        break;

    case 4:

        printf("Thursday");

        break;

    case 5:

        printf("Friday");

        break;

    case 6:

        printf("Saturday");

        break;

    case 7:

        printf("Sunday");

        break;

    default:

        printf("Invalid input! Please enter week number between 1-7.");

}

return 0;

}
```

Output:

```
/tmp/c1fos5GSxZ.o
13_Khushi Tiwari
Enter Day no between 1-7:6
Friday
```

Conclusion : Successfully performed a program to C program to print day name of week using switch-case

Experiment no – 03(c)

Aim: Write a program to read three values from keyboard and print out the largest of them without using if statement.

Algorithm:

- i. Ask the user to enter three integer values.
- ii. Read the three integer values in num1, num2, and num3 (integer variables).
- iii. Check if num1 is greater than num2.
- iv. If true, then check if num1 is greater than num3.
- v. If false, then check if num2 is greater than num3.

Code:

```
#include<stdio.h>

int main()
{ printf("13_Khushi Tiwari\n");
  int N1, N2, N3, Irg; printf("Enter three numbers:");
  scanf("%d %d %d", &N1, &N2, &N3);
  Irg = N1 > N2 ? (N1 > N3 ? N1 : N3) : (N2 > N3 ? N2 : N3);
  printf("%d is the largest number.",Irg);
  return 0;
}
```

Output:

```
/tmp/pzyoto5Uh3.o
13_Khushi Tiwari
Enter three numbers:4 5 6
6 is the largest number.
```

Conclusion : Successfully performed a program to read three values from keyboard and print out the largest of them without using if statement.

Experiment no – 04(a)

Aim: a. Write a program to print the pattern of asterisks as shown below :

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

Algorithm:

- i. Display * and go to new line
- ii. Display * * and go to new line.
- iii. Display * * * and go to new line.
- iv. Display * * * *

Code:

```
#include<stdio.h>  
  
int main()  
{ printf("13_Khushi Tiwari\n");  
  int i, j, n;  
  /*for used as row wise */  
  for(i=1; j<=4; ++i)  
  { /*for used as column wise */  
    for(j=1; j<=i; ++j)  
    {  
      printf("*"); } printf("\n");  
    }  
  return 0;}
```

Output:

```
/tmp/1hY6JsUCZX.o  
13_Khushi Tiwari  
Enter the number of rows: 5  
*  
* *  
* * *  
* * * *  
* * * * *
```

Conclusion : Successfully performed a program to print the pattern of asterisks.

Experiment no – 04(b)

Aim: Write a program to print the pattern of asterisks as shown below :

* * * * *

* * * *

* * *

* *

*

Algorithm:

- i. Display ***** and go to new line
- ii. Display * * **and go to new line.
- iii. Display * * * and go to new line.
- iv. Display * * and go to new line.
- v. Display *

Code:

```
#include<stdio.h>

int main()
{ printf("13_Khushi Tiwari\n");
  int i, j; /*for used as row wise */
  for(i=5; i>=1; i--)
  { /*for used as column wise */
    for(j=1; j<=i; j++) {
      printf("*"); }
    printf("\n"); }
  return 0; }
```

Output:

```
/tmp/NFUV0hXUNn.o
13_Khushi_Tiwari
Enter the number of rows: 5
* * * * *
* * * *
* * *
* *
*
```

Conclusion : Successfully performed a program to print the pattern of asterisks.

Experiment no – 04(c)**Aim: Write a program to print Floyd's Triangle.****Algorithm:**

- i. Create variables that hold rows and column values as i and j. Take a number to display the rows as num and set the variable k to 1 as its initial value.
- ii. Use nested for loops:
 - a. Outer for loop starts its iteration i = 1 up to n rows.
 - b. Inner for loop starts its iteration from j = 1 up to (j <=i).
- iii. Print the values of k.
- iv. Increment k by 1 or k = k + 1.
- v. Jump to newline after each iteration of the inner for loop.
- vi. Stop

Code:

```
#include <stdio.h>

int main()
{ printf("13_Khushi Tiwari\n");
  int n, i, c, a = 1;
  printf("Enter the number of rows of Floyd's triangle to print\n");
  scanf("%d", &n);

  for (i = 1; i <= n; i++)
  {
    for (c = 1; c <= i; c++)
    {
      printf("%d ", a); // Please note space after %d
      a++;
    }
    printf("\n");
  }
  return 0;
```

```
}
```

Output:

```
/tmp/BCCbpgbTc9.o
13_Khushi Tiwari
Enter a number to define the rows in Floyd's triangle:
5
1
  2  3
 4  5  6
 7  8  9 10|
11 12 13 14 15
```

Conclusion : Successfully performed a program to print Floyd's Triangle.

Experiment no – 05(a)

Aim: Write a program to print area of square using function.

Algorithm:

- i. Start.
- ii. Declare at s as integer.
- iii. Initialize value of s.
- iv. Calculate at $s \times s$.
- v. print area of triangle .
- vi. End.

Code:

```
#include <stdio.h>

int main()
{ printf("13_Khushi Tiwari\n");
  int s=13;
  int area_square=s*s;
  printf("Area of the square=%d",area_square);
}
```

Output:

```
/tmp/AgXStJtyHq.o
13_Khushi Tiwari
Enter side of square: 6
Area of Square: 36.000000
```

Conclusion : Successfully performed a program to print area of square using function.

Experiment no – 05(b)**Aim: Write a program using recursive function.****Algorithm:**

- i. Start.
- ii. Read the Input.
- iii. Perform recursion.
- iv. Print result.
- v. Stop.

Code:

```
#include <stdio.h>

int sum(int n);

int main() { printf("13_Khushi Tiwari\n");

    int number, result;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    result = sum(number);

    printf("sum = %d", result);

    return 0;

}

int sum(int n) {

    if (n != 0)

        // sum() function calls itself

        return n + sum(n-1);

    else

        return n;

}
```

Output:

```
/tmp/ZgWmARzfIg.o
13_Khushi Tiwari
Enter a positive integer :3|
Factorial of 3 is 6
```

Conclusion : Successfully performed a program using recursive function.

Experiment no – 05(c)**Aim: Write a program to square root, abs() value using function.****Algorithm:**

- i. Start
- ii. Read the input
- iii. Calculate absolute value
- iv. Calculate square root value
- v. Print results
- vi. Stop

Code:

```
#include<stdio.h>

#include<math.h>

int main()
{ printf("13_Khushi Tiwari\n");

  int num, a;

  printf("Please enter a number :\n");

  scanf("%d",&num);

  a = abs(num);

  printf("Calculated absolute value is : %d\n", a);

  a = sqrt(num);

  printf("Calculated Squareroot value is : %d\n",a);

  return 0;

}
```

Output:

```
/tmp/ZgWmARzfIg.o
13_Khushi Tiwari
Please enter a number:
4
Calculated absolute value is : 4
Calculated square root value is : 2
```

Conclusion : Successfully performed a program to square root, abs() value using function.

Experiment no – 05(d)**Aim: Write a program using go to statement.****Algorithm:**

- i. Start
- ii. Read the Input
- iii. Check if the input is inside loop or outside loop
- iv. Print result
- v. Stop

Code:

```
#include<stdio.h>

int main()
{ printf("13_Khushi Tiwari\n");
  int n;
  for(;;) /*infinite loop*/
  {
    printf("enter any number :");
    scanf("%d",&n);
    if(n == 5)
      goto ap; /* use of goto statement*/
    if (n% 2 == 0)
      continue; /*use of continue statement*/
    if (n% 3 == 0)
      break; /*use of break state*/
    printf("Inside loop");
  }
ap:
printf("Outside loop");
return 0;
}
```

Output:

```
/tmp/b5Ab5zIVcd.o
13_Khushi Tiwari
enter any number :13
Inside loopenter any number :13
Inside loopenter any number :13
Inside loopenter any number :
```

Conclusion : Successfully performed a program using go to statement.

Experiment no – 06(a)**Aim: a. Write a program to print rollno and names of 10 students using array.****Algorithm:**

- i. Start
- ii. Store Student Information
- iii. Create the student's structure variable
- iv. Display information
- v. Stop

Code:

```
// C Program to Store Information
// of Students Using Structure
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
// Create the student structure
struct Student {
    char* name;
    int roll_number;
};
// Driver code
int main()
{    printf("13_Khushi Tiwari\n");
    int i = 0, n = 10;
    // Create the student's structure variable
    // with n Student's records
    struct Student student[n];
    // Get the students data
    student[0].roll_number = 1;
    student[0].name = "Geeks16";
    student[1].roll_number = 2;
    student[1].name = "Geeks54";
    student[2].roll_number = 3;
```

```
student[2].name = "Geeks22";
student[3].roll_number = 4;
student[3].name = "Geeks41";
student[4].roll_number = 5;
student[4].name = "Geeks39";
student[5].roll_number = 6;
student[5].name = "Geeks3";
    student[6].roll_number = 7;
student[6].name = "Geeks32";
student[7].roll_number = 8;
student[7].name = "Geeks36";
    student[8].roll_number = 9;
student[8].name = "Geeks35";
student[9].roll_number = 10;
student[9].name = "Geeks34";
// Print the Students information
printf("Student Records:\n\n");
for (i = 0; i < n; i++) {
    printf("\tName = %s\n", student[i].name);
    printf("\tRoll Number = %d\n", student[i].roll_number);
}
return 0;
}
```

Output:

```
/tmp/b5Ab5zIVcd.o
13_Khushi Tiwari
Student Records:
|
    Name = Geeks16
    Roll Number = 1
```

Conclusion : Successfully performed a program to print rollno and names of 10 students using array.

Experiment no – 06(b)**Aim: Write a program to read a matrix of size m*n.****Algorithm:**

- i. Start
- ii. Enter row and column size
- iii. Construct Matrix
- iv. Display result
- v. Stop

Code:

```
#include<stdio.h>

int main()
{ printf("13_Khushi Tiwari\n");
  int i,j,m,n;
  float a[10][10];
  printf("Enter row and column size:\n");
  scanf("%d%d", &m, &n);
  printf("Enter matrix elements:\n");
  for(i=0;i< m;i++)
  {
    for(j=0;j< n;j++) {
      printf("a[%d][%d]=",i,j);
      scanf("%f", &a[i][j]);
    }
  }
  printf("Matrix read is:\n");
  for(i=0;i< m;i++)
  {
    for(j=0;j< n;j++)
    {
      printf("%f\t",a[i][j]);
    }
  }
  printf("\n");
```

```
}  
return 0;  
}
```

Output:

```
/tmp/b5Ab5zIVcd.o  
13_Khushi Tiwari  
Enter row and column size:  
2 2  
Enter matrix elements:  
a[0][0]=9  
a[0][1]=6  
a[1][0]=3  
a[1][1]=5  
Matrix read is:  
9.000000    6.000000  
3.000000    5.000000
```

Conclusion : Successfully performed a program to read a matrix of size $m*n$.

Experiment no – 06(c)

Aim: Write a program to sort the elements of array in ascending or descending order.

Algorithm:

- i. Start.
- ii. Input size of array.
- iii. Place currently selected element array to its correct place.
- iv. Swap if currently selected array element to its correct place.
- v. Print the sorted array.
- vi. Stop.

Code:

```
/**  
  
 * C program to sort elements of array in ascending order  
  
 */  
  
#include <stdio.h>  
  
#define MAX_SIZE 100 // Maximum array size  
  
int main()  
{ printf("13_Khushi Tiwari\n");  
  int arr[MAX_SIZE];  
  int size;  
  int i, j, temp;  
  /* Input size of array */  
  printf("Enter size of array: ");  
  scanf("%d", &size);  
  /* Input elements in array */  
  printf("Enter elements in array: ");  
  for(i=0; i<size; i++)  
  {  
    scanf("%d", &arr[i]);  
  }  
  
  for(i=0; i<size; i++)
```

```
{  
    /*  
    * Place currently selected element array[i]  
    * to its correct place.  
    */  
    for(j=i+1; j<size; j++)  
    {  
        /*  
        * Swap if currently selected array element  
        * is not at its correct position.  
        */  
        if(arr[i] > arr[j])  
        {  
            temp    = arr[i];  
            arr[i] = arr[j];  
            arr[j] = temp;  
        }  
    }  
}  
/* Print the sorted array */  
printf("\nElements of array in ascending order: ");  
for(i=0; i<size; i++)  
{  
    printf("%d\t", arr[i]);  
}  
return 0;  
}
```

Output:

```
/tmp/z6R9deSohX.o
13_Khushi Tiwari
Enter size of array: 4
Enter elements in array: 12 65 35 87
Elements of array in ascending order: 12    35  65  87
```

Conclusion : Successfully performed a program to sort the elements of array in ascending or descending order

Experiment no – 07(a)

Aim: Write a program to extract the portion of a character string and print the extracted part.

Algorithm:

- i. Start
- ii. Enter index start
- iii. Enter index last
- iv. Print result
- v. Stop

Code:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void demo(char*s,int start,int end)
{
    int i;
    for(i=start;i<=end;i++)
        printf("%c",s[i]);
}
int main()
{ printf("13_Khushi Tiwari\n");
char str[100]="NAVI MUMBAI";
int s,e;
printf("Enter the Start Index:");
scanf("%d",&s);
printf("Enter the Last Index:");
scanf("%d",&e);
if(e>strlen(str) ||(s>strlen(str)))
printf("The indeex's starting or ending value is out of range ");
else
demo(str,s,e);
return 0;
```

```
}
```

Output:

```
/tmp/z6R9deSohX.o
13_Khushi Tiwari
Enter the Start Index:3
Enter the Last Index:5
I M|
```

Conclusion : Successfully performed a program to extract the portion of a character string and print the extracted part.

Experiment no – 07(b)

Aim: Write a program to find the given string is palindrome or not.

Algorithm:

- i. Start
- ii. Check “hello” = palindrome
- iii. If true then print is a palindrome
- iv. If false then print is not a palindrome
- v. Check “madam”= palindrome
- vi. If true then print is a palindrome
- vii. If false then print is not a palindrome
- viii. Stop.

Code:

```
#include<stdio.h>
#include<string.h>
void isPal(char s[])
{
    int l = 0;
    int h = strlen(s)-1;
    while(h>l)
    {
        if(s[l++]!=s[h--])
        {
            printf("%s: not a palindrome\n",s);
            return;
        }
    }
    printf("%s :palindrome\n",s);
}
int main()
{
    printf("13_Khushi Tiwari\n");
    isPal("hello");
    isPal("madam");
    return 0;
}
```

```
}
```

Output:

```
/tmp/z6R9deSohX.o
13_Khushi Tiwari
hello: not a palindrome
madam : palindrome
```

Conclusion : Successfully performed a program to find the given string is palindrome or not.

Experiment no – 07(c)**Aim: Write a program to using strlen(), strcmp() function.****Algorithm:**

- i. Start
- ii. Use strlen function
- iii. Use strcmp function
- iv. Print result
- v. Stop

Code:

```
#include<stdio.h>
#include<string.h>
int main()
{ printf("13_Khushi Tiwari\n");
  int i;
  i=strlen("Hello");
  printf("\n %d",i);
  i=strcmp("Hello!","World");
  printf("\n %d",i);
  return 0;
}
```

Output:

```
/tmp/z6R9deSohX.o
13_Khushi Tiwari

5
-1
```

Conclusion : Successfully performed a program to using strlen(), strcmp() function.

Experiment no – 08(a)

Aim: Write a program to display the values using different data types and its address using pointer.

Algorithm:

- i. Start
- ii. Declare v1,v2,v3
- iii. Declare *p1,*p2,*p3
- iv. Insert values
- v. Print result
- vi. Stop

Code:

```
#include <stdio.h>

int main()
{ printf("13_Khushi Tiwari\n");
  int v1;
  float v2;
  char v3;
  int *p1;
  float *p2;
  char *p3;
  v1=11;
  v2=3.14;
  v3='Y';
  p1 = &v1;
  p2 = &v2;
  p3 = &v3;
  printf("Address of v1 = %u\n", &v1); printf("Value is = %d\n", *p1);
  printf("Address of v2 = %u\n", &v2);
  printf("Value is = %f\n", *p2);
  printf("Address of v3 = %u\n", &v3);
  printf("Value is = %c\n", *p3);
  return 0;
```

```
}
```

Output:

```
/tmp/z6R9deSohX.o  
13_Khushi Tiwari  
Address of v1 = 242079048  
Value is = 11  
Address of v2 = 242079052  
Value is = 3.140000  
Address of v3 = 242079047  
Value is = Y
```

Conclusion : Successfully performed a program to display the values using different data types and its address using pointer.

Experiment no – 08(b)**Aim: Write a program to perform addition and subtraction using pointer.****Algorithm:**

- i. Start
- ii. Enter numbers
- iii. Addition or Subtraction is performed
- iv. Display results
- v. Stop

Code:

```
//Add//

#include<stdio.h>

int main()
{ printf("13_Khushi Tiwari\n");
  int num1 ,num2, *p,*q,sum;
  printf("Enter any two integers:\n");
  scanf("%d%d", &num1,&num2);
  p = &num1;
  q = &num2;
  sum = *p+*q;

  printf("Sum= %d\n",sum);

  return 0;
}

//Sub//

#include<stdio.h>

int main()
{ printf("13_Khushi Tiwari\n");
  int num1 ,num2, *p,*q,sub;
```

```
printf("Enter any two integers:\n");  
scanf("%d%d", &num1,&num2);  
  
p = &num1;  
q = &num2;  
  
sub = *p-*q;  
  
printf("Sub= %d\n",sub);  
  
return 0;  
}
```

Output:

Add:

```
/tmp/z6R9deSohX.o  
13_Khushi Tiwari  
Enter any two integers:  
2 3  
Sum= 5
```

Sub:

```
/tmp/z6R9deSohX.o  
13_Khushi Tiwari  
Enter any two integers:  
5 2  
Sub= 3
```

Conclusion : Successfully performed a program to perform addition and subtraction using pointer.

Experiment no – 09(a)

Aim: Write a program to copy the contents of the file from one file into other.

Algorithm:

- i. Start
- ii. Create 2 files, f1 and f2
- iii. Add text to f1
- iv. Use getc, putc, FILE.
- v. Open f2
- vi. File f1's content has been copied to f2
- vii. Stop

Code:

```
#include<stdio.h>

main(){

    FILE *fp1, *fp2;

    char ch;

    fp1 = fopen("f1.txt", "r");
    fp2 = fopen("f2.txt", "w");

    while((ch = getc(fp1)) != EOF)

        putc(ch, fp2);

    fclose(fp1);
    fclose(fp2);

}
```

Output:**Conclusion :****Experiment no – 09(b)**

Aim: Write a program to print the structure using

- **Title**
- **Author**
- **Subject**
- **Book ID Print the details of two students.**

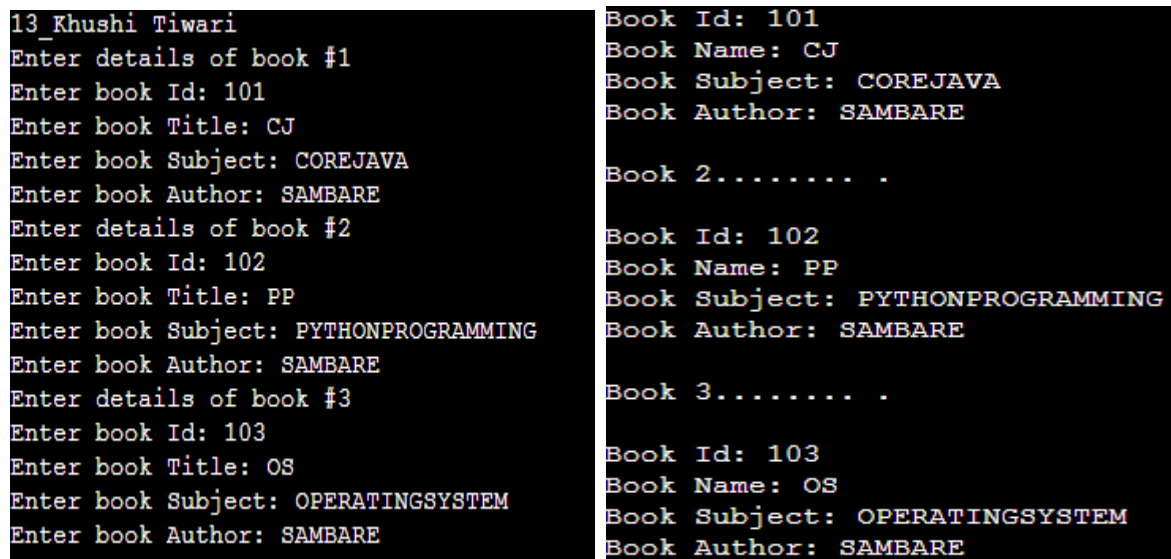
Code:

```
#include<stdio.h>

struct book{
char Title[40];
char Author[40];
char Subject[40];
int Book_ID;
};

int main() {
    printf("13_Khushi Tiwari\n");
    struct book b[3];
    int i;
    for(i=0; i<3; i++)
    {
        printf("Enter details of book #%d\n", i+1);
        printf("Enter book Id: ");
        scanf("%d", &b[i].Book_ID);
        printf("Enter book Title: ");
        scanf("%s", &b[i].Title);
        printf("Enter book Subject: ");
        scanf("%s", &b[i].Subject);
        printf("Enter book Author: ");
        scanf("%s", &b[i].Author); }
    for(i=0; i<3; i++)
    {
```

```
printf("\nBook %d..... .\n\n", i+1);  
printf("Book Id: %d\n", b[i].Book_ID);  
printf("Book Name: %s\n", b[i].Title);  
printf("Book Subject: %s\n", b[i].Subject);  
printf("Book Author: %s\n", b[i].Author); }  
return 0;  
}
```

Output:

The screenshot displays the output of a C program. It shows a user entering details for three books. The first book has ID 101, Name 'CJ', Subject 'COREJAVA', and Author 'SAMBARE'. The second book has ID 102, Name 'PP', Subject 'PYTHONPROGRAMMING', and Author 'SAMBARE'. The third book has ID 103, Name 'OS', Subject 'OPERATINGSYSTEM', and Author 'SAMBARE'. The output is formatted with color-coded text (green for IDs, red for names, blue for subjects, and yellow for authors) and includes a separator line 'Book 2..... .' between each book's details.

```
13_Khushi Tiwari  
Enter details of book #1  
Enter book Id: 101  
Enter book Title: CJ  
Enter book Subject: COREJAVA  
Enter book Author: SAMBARE  
Enter details of book #2  
Enter book Id: 102  
Enter book Title: PP  
Enter book Subject: PYTHONPROGRAMMING  
Enter book Author: SAMBARE  
Enter details of book #3  
Enter book Id: 103  
Enter book Title: OS  
Enter book Subject: OPERATINGSYSTEM  
Enter book Author: SAMBARE  
Book Id: 101  
Book Name: CJ  
Book Subject: COREJAVA  
Book Author: SAMBARE  
Book 2..... .  
Book Id: 102  
Book Name: PP  
Book Subject: PYTHONPROGRAMMING  
Book Author: SAMBARE  
Book 3..... .  
Book Id: 103  
Book Name: OS  
Book Subject: OPERATINGSYSTEM  
Book Author: SAMBARE
```

Conclusion : Successfully performed a program to print the details of two students.

Aim: Create a mini project on “Bank management system”. The program should be menu driven

Algorithm:

- i. Start
- ii. Enter number of customers record to enter
- iii. Read the number
- iv. Enter account number
- v. Enter name
- vi. Display Press 1 to deposit amount, Press 2 to withdraw amount, Press 0 to Exit.
- vii. Stop

Code:

```
#include <stdio.h>

struct customer {
int account_no;
char name[80];
int balance;
};

void accept(struct customer[], int);
int search(struct customer[], int, int);
void deposit(struct customer[], int, int, int);
void withdraw(struct customer[], int, int, int);

int main()
{
struct customer data[20];
int n, choice, account_no, amount, index;
printf("Banking System\n\n");
printf("Number of customer records you want to enter? :");
scanf("%d", &n);
accept(data, n);
do {
printf("\nBanking System Menu:\n");
printf("Press 1 to deposit amount.\n");
printf("Press 2 to withdraw amount.\n");
```



```
printf("Press 0 to exit\n");
printf("\nEnter choice(0-4): ");
scanf("%d", &choice);
switch (choice)
{
case 1:
printf("Enter account number: ");
scanf("%d", &account_no);
printf("Enter amount to deposit: ");
scanf("%d", &amount);
deposit(data, n, account_no, amount);
break;
case 2:
printf("Enter account number: ");
scanf("%d", &account_no);
printf("Enter amount to withdraw :");
scanf("%d", &amount);
withdraw(data, n, account_no, amount);
}
}
while (choice != 0);
return 0;
}
void accept(struct customer list[80], int s)
{
int i;
for (i = 0; i < s; i++)
{
printf("\nEnter data for Record #%d", i + 1);
printf("\nEnter account_no: ");
```

```
scanf("%d", &list[i].account_no);
printf("01-AlstonAlvares ");
gets(list[i].name);
list[i].balance = 0;
} }

int search(struct customer list[80], int s, int number)
{
    int i;
    for (i = 0; i < s; i++)
    {
        if (list[i].account_no == number) {
            return i;
        }
    }
    return -1; }

void deposit(struct customer list[], int s, int number, int amt)
{
    int i= search(list, s, number);
    if (i == -1) {
        printf("Record not found"); }
    else{
        list[i].balance+=amt;
    } }

void withdraw(struct customer list[], int s, int number, int amt)
{
    int i=search(list, s, number);
    if(i== -1)
    {
        printf("Record not found\n");
    }
    else if (list[i].balance < amt) {
```

```
printf("Insufficient balance\n"); }  
Else {  
list[i].balance-=amt;  
} }
```

Output:

Conclusion : Successfully performed a mini project on “Bank management system” . The program should be menu driven.