

#### 4(a) ADDITION OF TWO NUMBERS

**Aim:**

To write a C Program for addition of two numbers.

**Algorithm:**

STEP 1: Start the program.

STEP 2: Read the values of 'a' & 'b'.

STEP 3: Compute the sum of the entered numbers 'a', 'b',  $c=a+b$ .

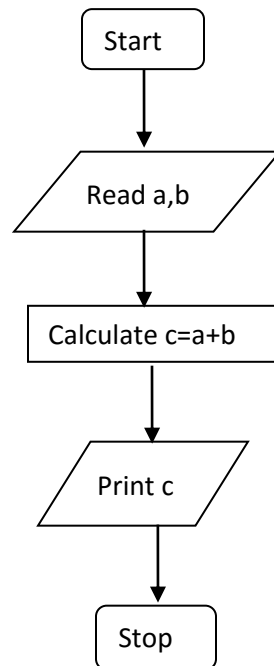
STEP 4: Print the value of 'c'.

STEP 5: Stop the program.

**Pseudo code:**

```
BEGIN  
READ 'a' & 'b'  
COMPUTE  $c=a+b$   
WRITE 'c'  
END
```

**Flow chart:**



**Result**

Thus the addition of two numbers program was successfully executed and verified.

#### 4(b) SWAPING OF TWO NUMBERS

##### Aim:

To write a C Program for swapping of two numbers by using arithmetic operator.

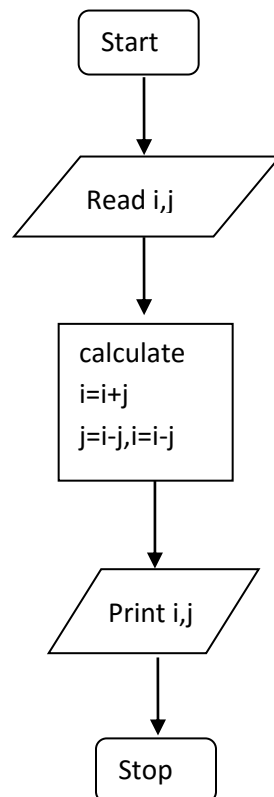
##### Algorithm

- Step 1: Start the program.
- Step 2: Declare the variable
- Step 3: swap the number using two variable.
- Step 4: Print the Results.
- Step 5: Stop the program.

##### Pseudocode:

```
BEGIN
READ
CALCULATE i=i+j
          j=i-j
          i=i-j
WRITE i,j
STOP
```

##### Flowchart:



**Result**

Thus the swapping of two numbers program was successfully executed and verified.

#### 4(c) TEMPERATURE CONVERSION

**Aim:**

To write a C Program for Celsius to Fahrenheit conversion.

**Algorithm**

STEP 1: Start the program.

STEP 2: Read the value of "Celsius "

STEP 3: Conversion of Celsius to Fahrenheit by using the formula

$$F = (1.8 * \text{Celsius}) + 32$$

STEP 4: Print the value of 'F'

STEP 5: Stop the program.

**Pseudo code:**

BEGIN

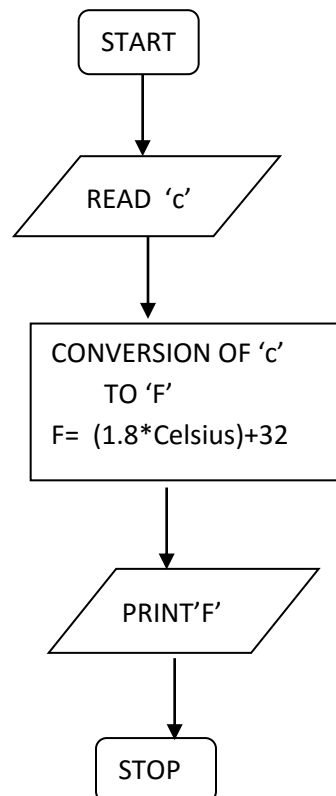
READ 'c'

COMPUTE  $F = (1.8 * \text{Celsius}) + 32$

WRITE 'F'

END

**Flow chart:**



**Result:**

Thus the Temperature conversion program was successfully executed and verified.

#### 4(d) QUADRATIC EQUATION

##### Aim:

To write a C program for determine roots of a real type quadratic equation  $ax^2 + bx + c = 0$ .

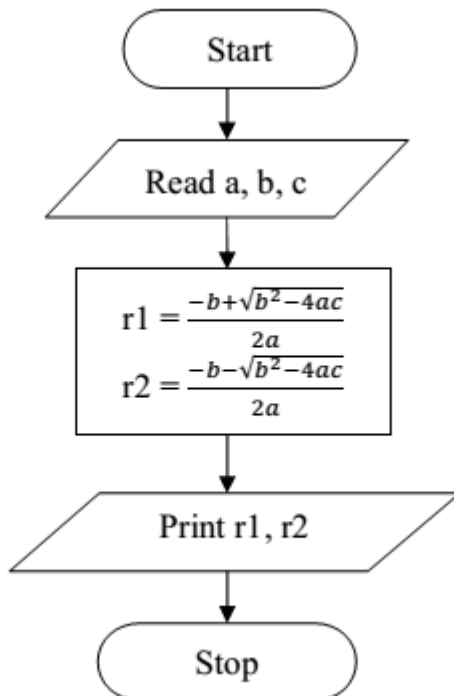
##### Algorithm:

- Step 1 : Start the program.
- Step 2 : Read co-efficient values of a,b and c
- Step 3: Compute roots  $r1 = (-b + \sqrt{d}) / (2.0 * a);$   
 $r2 = (-b - \sqrt{d}) / (2.0 * a);$
- Step 4 : Print r1, r2
- Step 5 : Stop the program.

##### Pseudo code:

```
BEGIN
READ a,b,c
COMPUTE roots  $r1 = (-b + \sqrt{d}) / (2.0 * a);$ 
                $r2 = (-b - \sqrt{d}) / (2.0 * a);$ 
WRITE r1,r2
TERMINATE
```

##### Flow chart:



**Result:**

Thus the roots of a quadratic equation were obtained using math library functions.



#### 4(e) FINDING THE AREA, CIRCUMFERENCE OF CIRCLE

**Aim:**

To write a C program for compute area and circumference of a circle.

**Algorithm:**

- Step 1 : Start the program
- Step 2 : Define constant  $\pi = 3.14$
- Step 3 : Read the value of radius
- Step 4 : Calculate area using formulae  $\pi * \text{radius}^2$
- Step 5 : Calculate circumference using formulae  $2 * \pi * \text{radius}$
- Step 6 : Print area and circumference
- Step 7 : Stop the program

**Pseudo code:**

BEGIN

READ r

COMPUTE

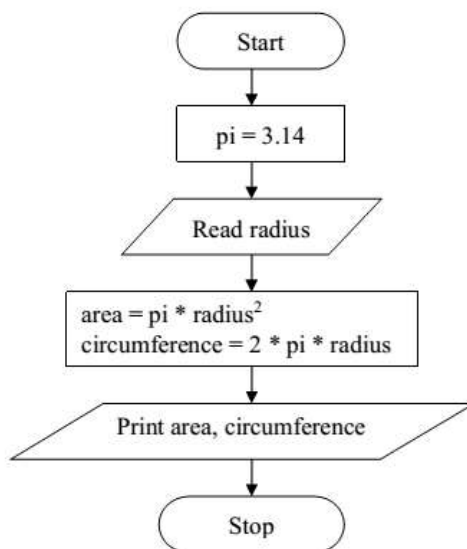
Area=  $\pi * \text{radius}^2$

Circumference=  $2 * \pi * \text{radius}$

WRITE area, circumference

END

**Flowchart:**



**Result:**

Thus the area and circumference of a circle program was successfully executed and verified.

#### 4(f) LEAP YEAR OR NOT

##### Aim:

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

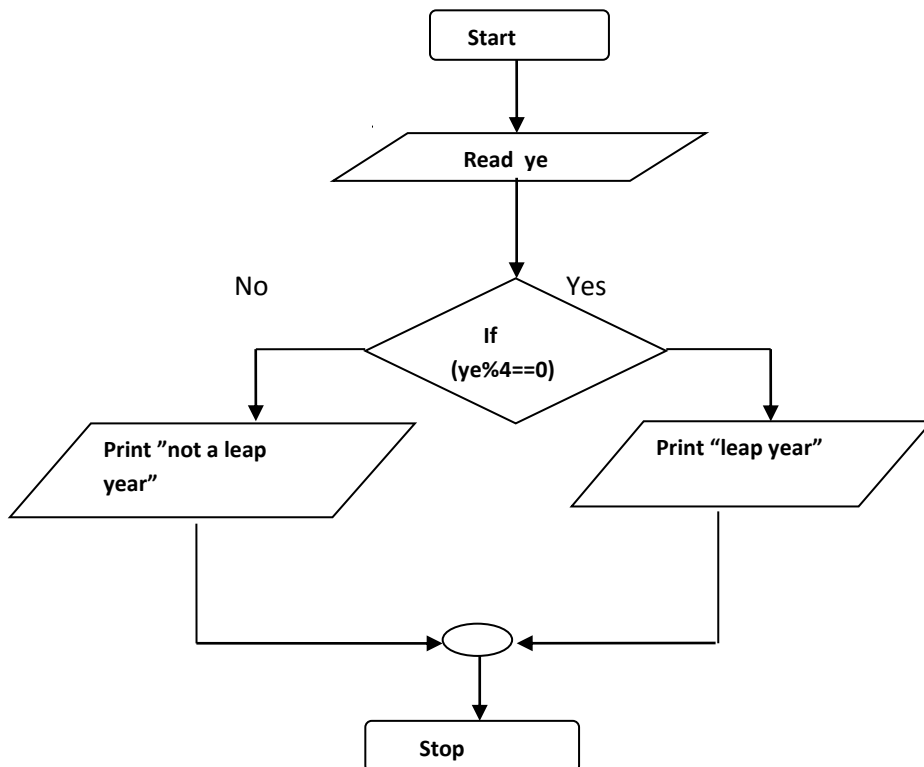
##### Algorithm

Step1: Start the program  
Step2: Declare year as int data type  
Step3: Read the year  
Step4: Check if  $(ye \% 4 == 0)$   
    STEP4.1: Print "It is a leap year"  
Step5: Else  
    Step5.1: Print "It is not a leap year"  
Step6: stop the program.

##### Pseudo code

```
Set initial year.  
READ the year.  
IF  $(ye \% 4 == 0)$  THEN  
    WRITE the year is leap year.  
ELSE  
    WRITE the year is not a leap year.  
ENDIF  
END
```

##### Flow Chart



**Result**

Thus the leap year program was successfully executed and verified.

4(g) ODD OR EVEN

**Aim**

To write a C program to find whether the given number is even or odd.

**Algorithm**

Step-1: Start the program.

Step-2: Read num

Step-3: If the number is divided by 2 , print the given number is even .

Step-4: If the number is not divided by 2, print the given number is odd.

Step-5: Stop the program.

**Pseudo code**

BEGIN

READ num

IF (num%2==0)

    WRITE even

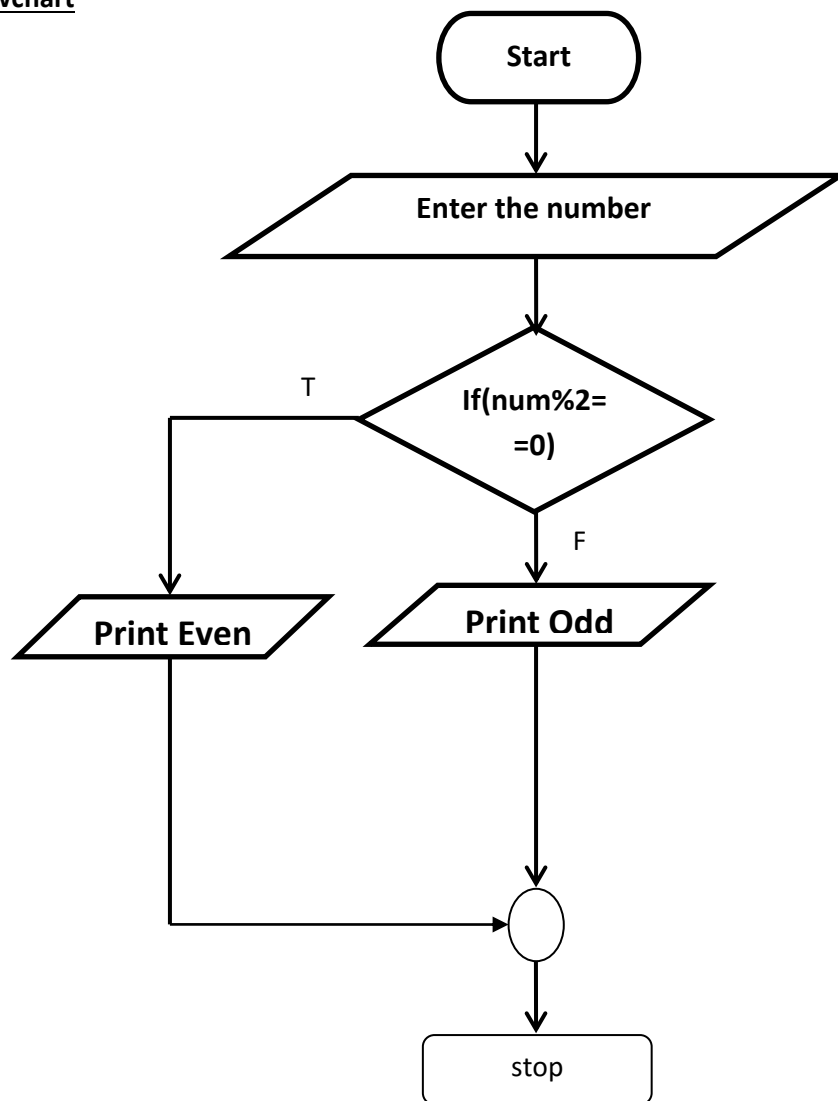
ELSE

    WRITE odd

ENDIF

END

## Flowchart



**Result**

Thus the odd or even program was successfully executed and verified.

#### 4(h) GREATEST OF TWO NUMBERS BY USING TERNARY OPERATOR

**Aim:**

To write a C program to find the biggest of two numbers using ternary operator.

**Algorithm:**

Step 1 : Start the program

Step 2 : Read the input value of X,Y

Step 3: Compute biggest among two numbers X,Y and print it

Step 4 : Stop the program

**Pseudo code:**

BEGIN

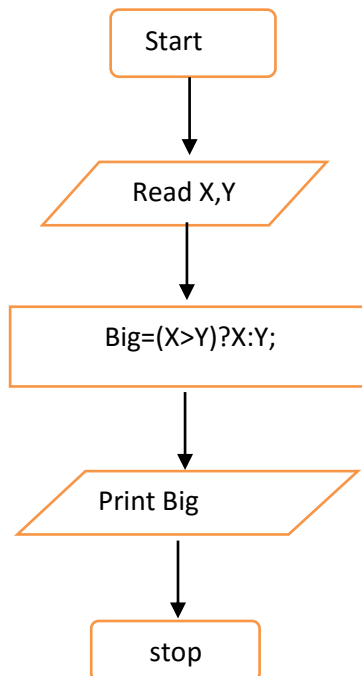
READ X,Y

COMPUTE big=(X>Y)?X:Y;

WRITE BIG

STOP

**FLOWCHART:**





**Result:**

Thus the biggest of two numbers program was successfully executed and verified

5(a)    GREATEST OF TWO NUMBERS BY USING IF STATEMENT

**Aim:**

To write a C Program to find greatest among two numbers using if else statement.

**Algorithm:**

STEP 1: Start the program

STEP 2: Read the values of 'a' & 'b'

STEP 3: Compare the greatest among the two values

STEP 4: Print the greatest number

STEP 5: Stop the program

**Result**

Thus the greatest of two number program was successfully executed and verified.

5(b)     GREATEST AMONG THREE NUMBERS USING NESTED IF STATEMENT

**Aim :**

To write a C Program to find the greatest among three numbers.

**Algorithm:**

STEP 1:Start the program

STEP 2:Read the values of 'a','b'&'c'.

STEP 3:Compare the greatest among the three values

STEP 4:Print the greatest number

STEP 5:Stop the program

**Result**

Thus the greatest among three numbers program was successfully executed and verified.

## 5(C) CALCULATOR USING SWITCH CASE

Aim:

To write C Program for simple calculator using switch case statement.

Algorithm:

STEP 1: Start the program

STEP 2: Read the values of 'a' & 'b'

STEP 3: Read the choice from 1 to 4

STEP 4: If the choice is 1 means then go to perform addition and print result else

STEP 5: If the choice is 2 means then go to perform subtraction and print result else

STEP 6: If the choice is 3 means then go to perform multiplication and print result else

STEP 7: If the choice is 4 means then go to perform division and print result else

STEP 8: go to default statement

STEP 5: Stop the program

### **Result**

Thus the calculator program was successfully executed and verified.

5(d) FIND THE SUM OF DIGITS

**Aim:**

To write a C Program to find the sum of digits of given number.

**Algorithm:**

STEP 1: Start the program.

STEP 2: Read the value of "r"

STEP 3: Compute the entered digits WHILE( $n > 0$ )

$x = n \% 10$

$S = S + x$

$N = n / 10$

STEP 4: Print the value of 's'

STEP 5: Stop the program.



### **Result**

Thus the sum of digits program was successfully executed and verified.

5(e) FINDING WHETHER A GIVEN NUMBER IS ARMSTRONG OR NOT

**Aim:**

To write a C program to find the given number is Armstrong or not.

**Algorithm:**

STEP 1: Start the program.

STEP 2: Declare n and sum

STEP 3: Read the value of n

STEP 4: till the a is not equal to 0 the following step is executed

STEP 5: Compute the following

$a = n \% 10$

$b = b + a * a * a$

$n = n / 10$

STEP 6: Print the answer whether it is an Armstrong or not

STEP 7: Stop the program.

## **Result**

Thus the Armstrong or not program was successfully executed and verified.

5(f) FINDING PALINDROME OR NOT

**Aim:**

To write C Program to find whether a given number is palindrome or not.

**Algorithm:**

STEP 1: Start the program.

STEP 2: Declare n, a, temp and rev

STEP 3: Read the value of n

STEP 4: Compute the following

$a = n \% 10$

$n = n / 10$

$rev = rev * 10 + a$

STEP 5: If (temp=rev) follow step 6

STEP 6: Print the answer whether it is a palindrome or not

STEP 7: Stop the program.

**Result**

Thus the given number is palindrome or not program is successfully executed and verified.

5(g) PRIME NUMBER

**Aim:**

To write a C program whether the given number is prime or not

**Algorithm:**

Step 1: Start the program

Step 2: Read the values of the n

Step 3: check the IF statement `if(n%i==0)`

Step 4: The looping statement is executed [`for(i=2; i<=n/2; i++)`]

Step 5: print prime or not.

Step 6: Stop the program.

**Result**

Thus the given number is prime or not program was successfully executed and verified.

5(h) FIBBANOCCHI SERIES

**Aim**

To 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)



**Result**

Thus the Fibonacci series program was successfully executed and verified.

**5(i)** FACTORIAL

**Aim:**

To write a C program for finding the factorial of given number.

**Algorithm:**

Step 1: Start the program

Step 2: Read the n value

Step 3: Execute the for looping until the entered n

Step 4: compute  $\text{fact} = \text{fact} * i$

Step 5: Print the factorial for the entered number

Step 6: Stop the program

### **Result**

Thus the factorial program was successfully executed and verified.

5(j) THE NUMBER DIVISIBLE BY 2 BUT NOT DIVISIBLE By 3 AND 5 UPTO 100

**Aim:**

To write the C program for the number divisible by 2 but not divisible by 3 and 5 upto 100

**Algorithm:**

Step 1:Start the program

Step 2:Use the for(a=0;a<100;a++)

Step 3:Check the if condition

(a%2==0&&a%3!=0&&a%5!=0)

Step 4: print the number divisible by 2

Step 5: Stop the program.

**Result**

Thus the program was successfully executed and verified.

## 6(a) SORTING

### **Aim:**

To write a C program for sorting of elements by using one dimensional array

### **Algorithm**

- Step 1.start the program
- STEP2. Read the number of values (n).
- STEP 3.Initialize the iterative value (i) to 0.
- STEP 4. Repeat the steps 4 to 5 until the number of values minus 1 (n-1).
- STEP 5. Read the elements in an array a[i].
- STEP 6.Increment the iterative value.
- STEP 7.Initialize the iterative value (i) to 0.
- STEP 8. Repeat the steps 8 to 12 until the number of values minus 1 (n-1).
- STEP 9. Assign the iterative value j as i+1.
- STEP 10. Repeat the steps 10-11 until iterative value (j) less than no. of values(n).
- STEP 11.If  $a[i] > a[j]$  then
  - Swap the values using temporary variable.
- STEP 12.Increment the iterative value (j)
- STEP 13.Increment the iterative value (i)
- STEP 14. Display the array elements in the ascending and descending order
- Step 15.stop the program.

**Result**

Thus the sorting program was successfully executed and verified.

## **6(b)    SEARCHING**

### **Aim:**

To write a C program for searching the element by using one dimensional array

### **Algorithm:**

Start the program

1. Read the number of values (n).
  2. Initialize the iterative variable  $i=0$ .
  3. Repeat the steps 4&5 until no. of values minus 1 ( $n-1$ ).
  4. Read the elements in an array.
  5. Increment the iterative variable.
  6. Assign the first element of an array to the small and big variables.
  7. Repeat the steps 8 to 10, until no of values minus 1 ( $n-1$ ).
  8. If the array element is greater than big element, Assign the element to the big variable.  
Otherwise go to step 10.
  9. If the array element is smaller than small element, Assign the element to the small variable.  
Otherwise go to step 10.
  10. Increment the iterative variable.
  11. Display the biggest element in the list.
  12. Display the smallest element in the list.
- Stop the program



### **Result**

Thus the searching program was successfully executed and verified.

### **6(c). MATRIX ADDITION**

**Aim:**

To write a C program for matrix addition by using two dimensional array

**Algorithm:**

Start the program

1. Read the total number of rows & columns for the matrix
  2. Read the elements for A Matrix and store it in to two dimensional arrays.
  3. Read the elements for B Matrix and store it into other two dimensional array.
  4. Add the elements of 2 matrixes namely A & B through the nested for Loops and store the result into a resultant 2-dimesional array C.
  5. Display the result
- Stop the program.

**Result**

Thus the matrix addition program was successfully executed and verified.

## **6(d)     MATRIX MULTIPLICATION**

### **Aim:**

To write a C program for matrix multiplication by using two dimensional array

### **Algorithm:**

Start the program

1. Read the total number of rows & columns for the matrix
  2. Read the elements for A Matrix and store it into two dimensional arrays.
  3. Read the elements for B Matrix and store it into other two dimensional array.
  4. Multiply the elements of 2 matrixes namely A & B through the nested for Loops and store the result into a resultant 2-dimesional array C.
  5. Display the result
- Stop the program

**Result**

Thus the matrix multiplication program was successfully executed and verified.

## **6(e)     MATRIX TRANSPOSE**

### **Aim:**

To write C program for Transpose of matrix by using two dimensional array

### **Algorithm:**

Start the program

1. Read the total number of rows & columns for the matrix
2. Read the elements for A Matrix and store it into two dimensional arrays.
3. Transpose of matrix A is computed
4. display the result.

Stop the program.

**Result**

Thus the transpose of matrix program was successfully executed and verified.

## **7(A) STRING CONCATENATION**

### **Aim:**

To write C program for string concatenation using string function.

### **Algorithm:**

Start the program

1. Read two Strings (str1,str2).
2. Copy each character from string1 to another string(str3) until the null character of string1 is reached.
3. Continuously copy each character from string2 to string(str3)until the null characterof string2 is reached.
4. Add the null character to the end of the String(str3).
5. Display the concatenated String(str3).

Stop the program.



**Result**

Thus the concatenation program was successfully executed and verified.

## **7(b) STRING PALINDROME USING STRING FUNCTION**

### **Aim:**

To write a C program for string palindrome using string function

### **Algorithm:**

STEP1: Start the program

STEP2: Read the character input of string1 variable

STEP3: Assign the value of string1 in to string2 variable

STEP4: find the reverse of string2 using string function `strrev()`

STEP5: compare the string1 and string2 if return value true means print palindrome

STEP6: otherwise print not palindrome

STEP7: stop the program

**Result**

Thus the string palindrome program was successfully executed and verified

### **8(a) SWAPING OF TWO NUMBERS USING CALL BY VALUE**

**Aim:**

To write C Program for swapping of two numbers using call by value function

**Algorithm:**

STEP1: Start the program

STEP2: Read the input value 'a' & 'b'

STEP3: Call function swap(a,b)

STEP4: print the value of 'a' and 'b'

STEP5: stop the program

STEP1: function swap start

STEP2: swap two numbers using temporary variable

STEP3: return swap value to main function

**Result**

Thus the swapping program was successfully executed and verified

## **8(b) SWAPING OF TWO NUMBERS USING CALL BY REFERENCE**

### **Aim:**

To write a C Program for swapping of two numbers using call by reference

### **Algorithm:**

STEP1: Start the program

STEP2: Read the input value 'a' & 'b'

STEP3: Call function swap(&a,&b)

STEP4: print the value of 'a' and 'b'

STEP5: stop the program

STEP1: function swap start

STEP2: swap two numbers using temporary variable and pointer variable

STEP3: return swap value to main function

**Result**

Thus the swapping program was successfully executed and verified

## 9. FACTORIAL USING RECURSION

### Aim:

To write a C program to calculate the factorial of the given number using recursive functions.

### Algorithm:

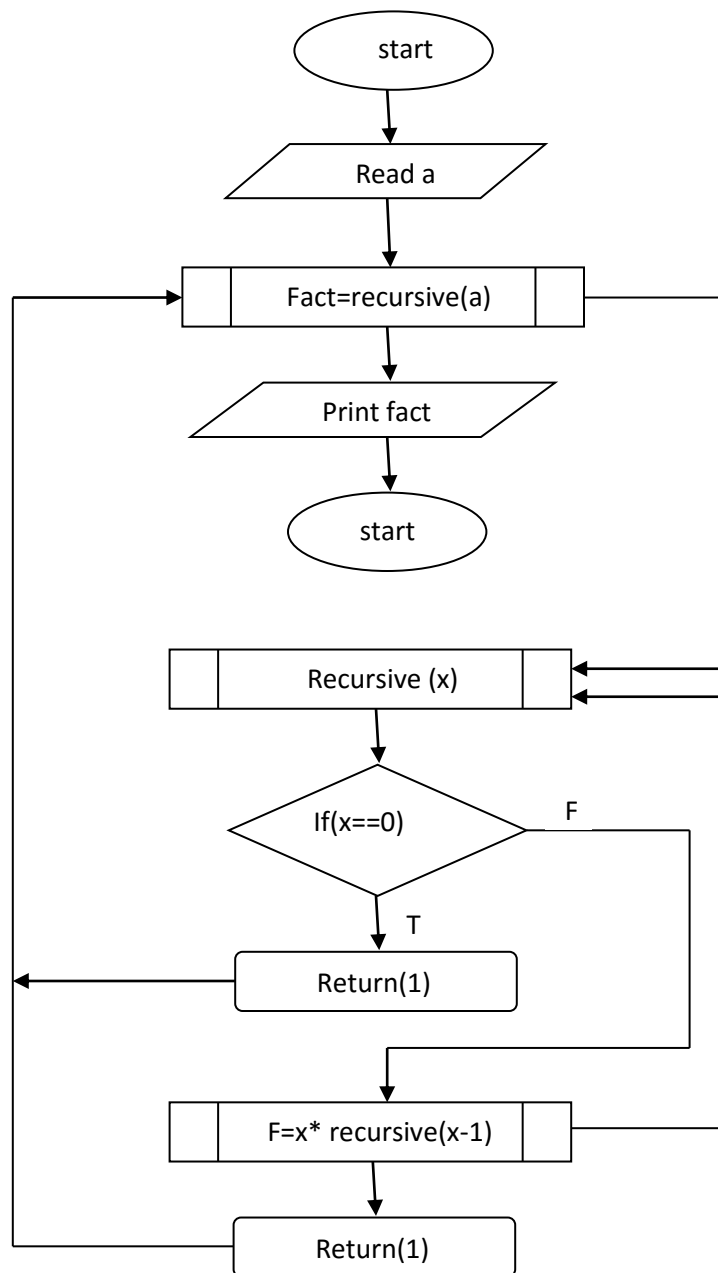
Start the program

1. Read the number (a)
2. Call the function rec() by passing the number(a)
3. If the number is 0 Return the factorial value as 1.
4. Else Compute the factorial value by recursively calling rec() function by decrementing the number(x).
5. Return the factorial value(f).
6. Display the factorial value.

Stop the program.



Flow chart:



## **Result**

Thus the factorial program was successfully executed and verified

## 10(a) EMPLOYEMENT PAYROLL USING STRUCTURE

### Aim:

To write a C program to print employee details using structure

### Algorithm:

STEP1: start the program

STEP2: Define structure employee with the following members

```
char name[15];
```

```
int empid;
```

```
float bsal;
```

```
float nsal;
```

```
float gross;
```

STEP3: Read employee details of name, id and basic salary

STEP4: Calculate

```
hra=((10*emp.bsal)/100);
```

```
da=((35*emp.bsal)/100);
```

```
tax=((15*emp.bsal)/100);
```

```
gross=bsal+hra+da;
```

```
net=gross-tax;
```

STEP5: Display the employee details with hra, da, tax, gross and net salary

STEP6: stop the program

RESULT:

Thus the employee payroll program was successfully executed and verified



## 10(b) STUDENT MARKLIST USING UNION

Aim:

To write a C program to print student grade using union.

Algorithm:

Step 1: start the program.

Step 2: define union student with the following members

char name[20];

char regno[12];

int avg;

char grade;

step 3: read the student details and marks

step 4: calculate average and grade

step 5: display the student details with average and grade.

Step 6: stop the program.



Result:

Thus the student mark list program is successfully executed and verified.

