**SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA**

**(UNIVERSITY ESTABLISHED under section 3 of UGC Act 1956)**

**ENATHUR,** **KANCHIPURAM – 631 561**



**PROBLEM SOLVING TECHNIQUES LAB**

**LABORATORY RECORD**

**Name :** T. ROHITH

**Reg. No :** 112534041

**Class :** I YEAR BCA

**Subject :** UCAF251T50 - PST IN C LAB

**SRI CHANDRASEKHARENDRA SARASWATHI**

**VISWA MAHAVIDYALAYA**

**(University Established under section 3 of UGC Act 1956)**

****

**BONAFIDE CERTIFICATE**

**This is to Certify that this is the bonafide record of work done by Mr. T. ROHITH**, **with Reg. No 112534041 of I Year BCA in the Problem Solving Techniques Lab during the year 2025.**

**Staff-in-charge** **Head of the Department**

**Submitted for the Practical Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Internal Examiner** **External Examiner.**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Date** | **Title** | **Page No.** | **Signature** |
| 1 | 24 Jul 2025 | [CELSIUS TO FAHRENHEIT](#CELSIUS) | 1 |  |
| 2 | 31 Jul 2025 | [FIBONNACI SERIES](#SERIES) | 4 |  |
| 3 | 07 Aug 2025 | [SQUARE AND CUBE OF NUMBERS](#squarecube) | 7 |  |
| 4 | 14 Aug 2025 | [GENERATE ODD NUMBERS](#Odd) | 10 |  |
| 5 | 21 Aug 2025 | [GENERATE GRADE](#grade) | 13 |  |
| 6 | 28 Aug 2025 | [GENERATE TOWER](#tower) | 16 |  |
| 7 | 04 Sep 2025 | [PALINDROME OR NOT](#palindrome) | 19 |  |
| 8 | 25 Sep 2025 | [STRING HANDLING](#string) | 22 |  |
| 9 | 25 Sep 2025 | [ARRAY SORTING](#array) | 26 |  |
| 10 | 09 Oct 2025 | [FACTORIAL](#factorial) | 29 |  |
| 11 | 09 Oct 2025 | [SWAPPING OF NUMBER](#swapping) | 31 |  |
| 12 | 23 Oct 2025 | [USING STRUCTURES](#structure) | 33 |  |
| 13 | 23 Oct 2025 | [FILE PROGRAMMING](#file) | 36 |  |

|  |  |  |
| --- | --- | --- |
| 1 | **CELSIUS TO FAHRENHEIT** | DATE: 24 Jul 2025 |

**AIM:**

Write a C program to Convert temperature from Celsius to Fahrenheit and vice versa.

**ALGORITHM:**

**Step 1:** Start the program.

**Step 2:** Display the menu options:

* Option 1 → Convert Celsius to Fahrenheit
* Option 2 → Convert Fahrenheit to Celsius

**Step 3:** Read the user’s choice.

**Step 4:** if the choice = 1 goto step 5, else goto step 9

**Step 5:** Read the Celsius value.

**Step 6:** Convert Celsius to Fahrenheit using the formula: Fahrenheit=(Celsius×9/5) +32

**Step 7:** display the converted Fahrenheit value.

**Step 8:** Goto step 15.

**Step 9:** if the choice is 2: goto step 11. Else goto step 14.

**Step 11:** Read the Fahrenheit value.

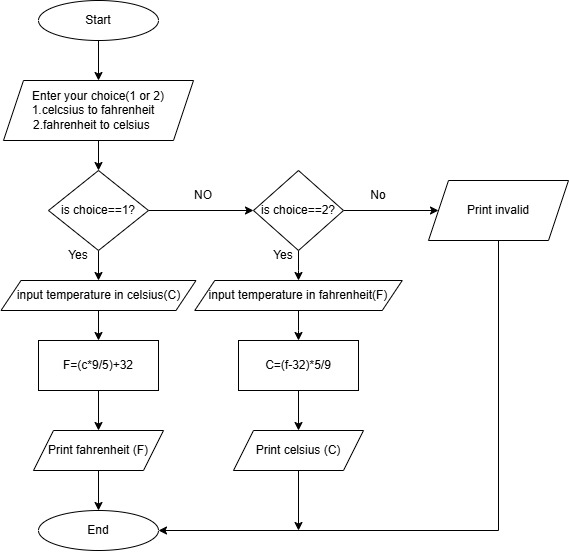
**Step 12:** Convert it to Celsius using the formula: Celsius=(Fahrenheit−32) ×5/9.

**Step 13:** Display the converted Celsius value.

**Step 14:** Display an error message: “Invalid choice! Please run the program again and choose 1 or   
 2.”

**Step 15:** End the program.

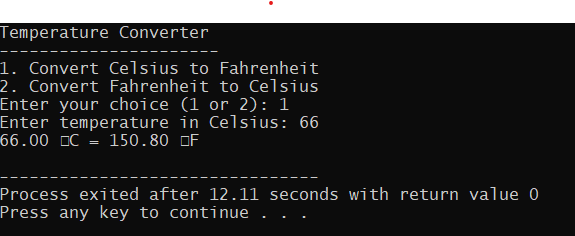
**FLOW CHART:**

****

**SOURCE CODE:**

<https://github.com/murugamuruga123/clab/blob/main/CELSIUS%20TO%20FAHRENHEIT.c>

**OUTPUT:**

****

**RESULT:**

Thus the program is compiled and executed successfully with verified output.

|  |  |  |
| --- | --- | --- |
| 2 | **FIBONACCI SERIES** | DATE: 31 Jul 2025 |

**AIM:**

Write a C program to generate a Fibonacci series.

**ALGORITHM:**

**Step 1:** Start the program.

**Step 2:** Declare variables:  
  n (number of terms),  
  t1 = 0 (first term),  
  t2 = 1 (second term),  
  next Term (to store next Fibonacci number),  
  i (loop counter).

**Step 3:** read the number of terms n.

**Step 4:** i=3

**Step 5**: Calculate next Term = t1 + t2.

**Step 6**: Print next Term.

**Step 7:** t1 = t2,

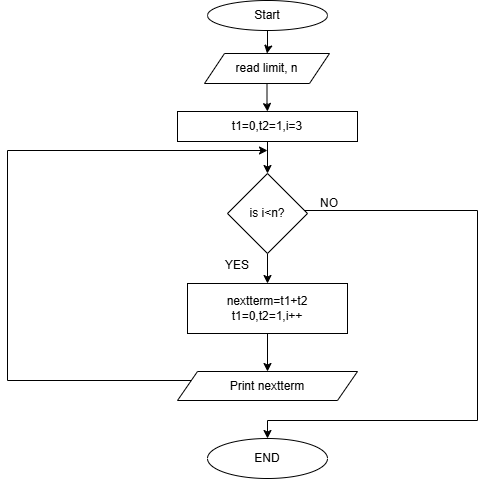
t2 = next Term.

i=i+1

**Step 8**: if i<n repeat step 5, else continue

**step 9:** End the program.

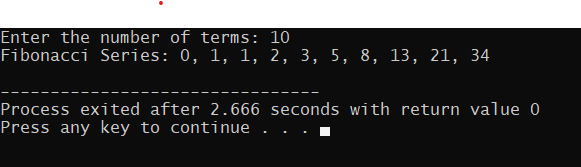
**FLOW CHART:**



**SOURCE CODE:**

[**https://github.com/murugamuruga123/clab/blob/main/fibonnaci%20series.c**](https://github.com/murugamuruga123/clab/blob/main/fibonnaci%20series.c)

**OUTPUT:**

****

**RESULT:**

Thus the program is compiled and executed successfully with verified output.

|  |  |  |
| --- | --- | --- |
| 3 | **SQUARE AND CUBE OF NUMBERS** | DATE:07 Aug 2025 |

**AIM:**

Write a C program to calculate the square and cube of 1 to n numbers

**ALGORITHM:**

**Step 1:** Start

**Step 2:** Declare integer variables limit(n) and counter (i)

**Step 3:** Read the value n

**Step 4:** Use a **for loop** that runs from i = 1

**Step 5:** Calculate square = i \* i

**Step 6:** Calculate cube = i \* i \* i

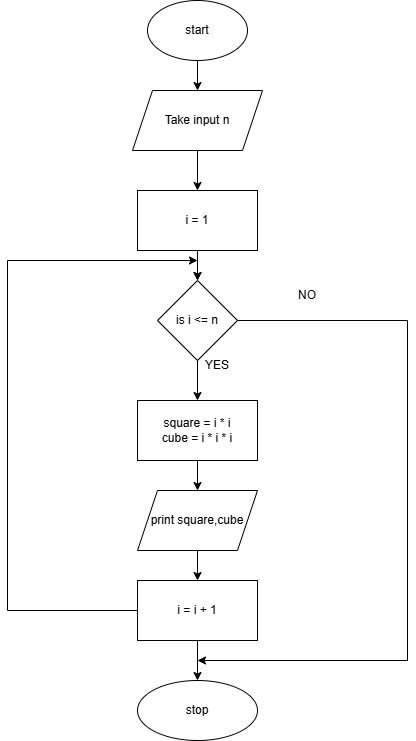
**Step 7:** Display i, square, and cube

**Step 8:** i = i + 1

**Step 9:** If i < n go to step 5, else go to step 10

**Step 10:** stop

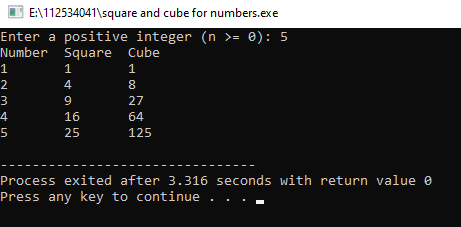
**FLOWCHART:**



**SOURCE CODE :**

<https://github.com/Rohith-Tech-king/c-lab/blob/main/square%20and%20cube%20for%20numbers.c>

**OUTPUT:**



**RESULT:**

Thus the program is compiled and executed successfully with verified output.

|  |  |  |
| --- | --- | --- |
| 4 | **GENERATE ODD NUMBERS** | DATE:14 Aug 2025 |

**AIM:**

Write a C program to display odd numbers from 1 to n

**ALGORITHM:**

**Step 1:** Start

**Step 2:** Declare integer variables limit (n) and counter (i)

**Step 3:** Read the value of n

**Step 4:** Use a for loop that runs from i = 1

**Step 5:** if i % 2! = 0 (i.e., i is odd) go to step 6, else go to step 7

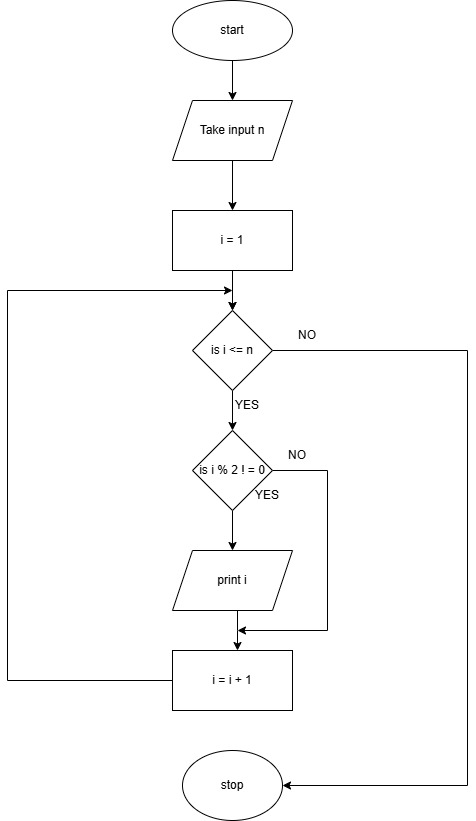
**Step 6:** Print i

**Step 7:** i=i+1

**Step 8:** if i<n Go to Step 5, else goto step 9

**Step 9:** Stop

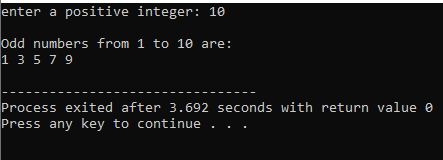
**FLOWCHART:**



**SOURCE CODE:**

[**https://github.com/Rohith-Tech-king/c-lab/blob/main/odd%20num.c**](https://github.com/Rohith-Tech-king/c-lab/blob/main/odd%20num.c)

**OUTPUT:**

****

**RESULT:**

Thus the program is compiled and executed successfully with verified output**.**

|  |  |  |
| --- | --- | --- |
| 5 | **GENERATE GRADE** | DATE:21 Aug 2025 |

**AIM:**

To write a C program that reads marks (from 0 to 100) and prints the corresponding grade using a switch-case statement based on this grading:

**ALGORITHM:**

**Step 1:** start

**Step 2**: Declare a variable(n) to store marks

**Step 3:** enter marks

**Step 4:** Check if 0 <= marks <= 100 continue else go to step 8

**Step 5:** calculate range=marks/10

**Step 6:** Use switch case to decide the grade:

If range (80-100) grade= A

If range (60-79) grade =B

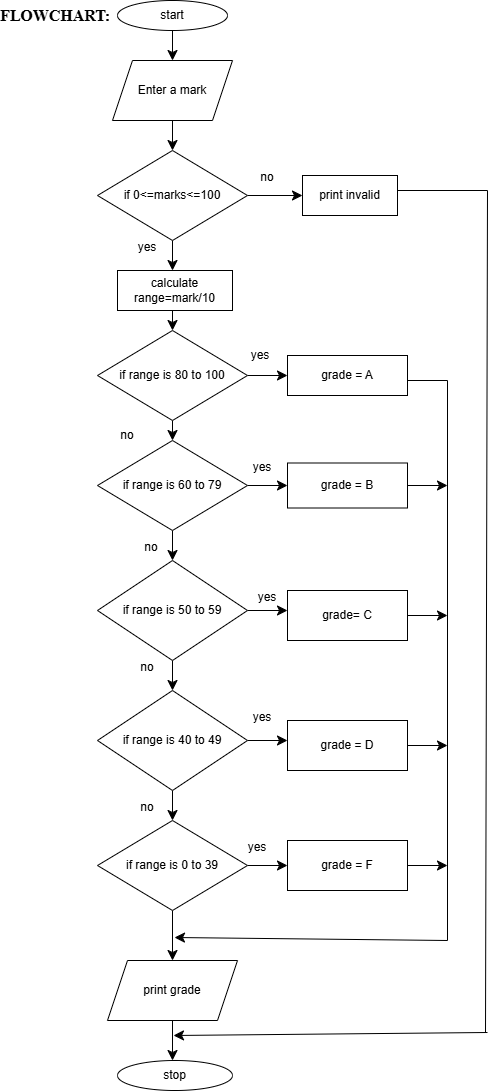
If range (50-59) grade =C

If range (40-49) grade =D

If range (0-39) grade=F

**Step 7:** Print the grade.

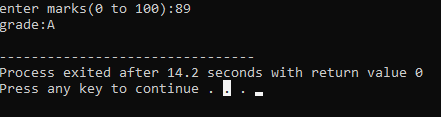
**Step 8:** End



**SOURCE CODE:**

[**https://github.com/sruthisruthi18112007/c-lab/blob/main/grade.c**](https://github.com/sruthisruthi18112007/c-lab/blob/main/grade.c)

**OUTPUT:**

****

**RESULT:**

Thus the program is compiled and executed successfully with verified output.

|  |  |  |
| --- | --- | --- |
| 6 | **GENERATE TOWER** | DATE:28 Aug 2025 |

**AIM:**

To write a C program that prints a right angled triangle pattern using a given character and number of lines.

**ALGORITHM:**

Step 1:start.

Step 2:declare variables: Limit (n), counters (i, j).

Step 3**:** enter a character to print**,** ch.

Step 4: enter a limit n.

Step 5:i=1.

Step 6: j=1.

Step 7: Print character.

Step 8**:** j++.

Step 9: If j<i, goto step 7, else continue

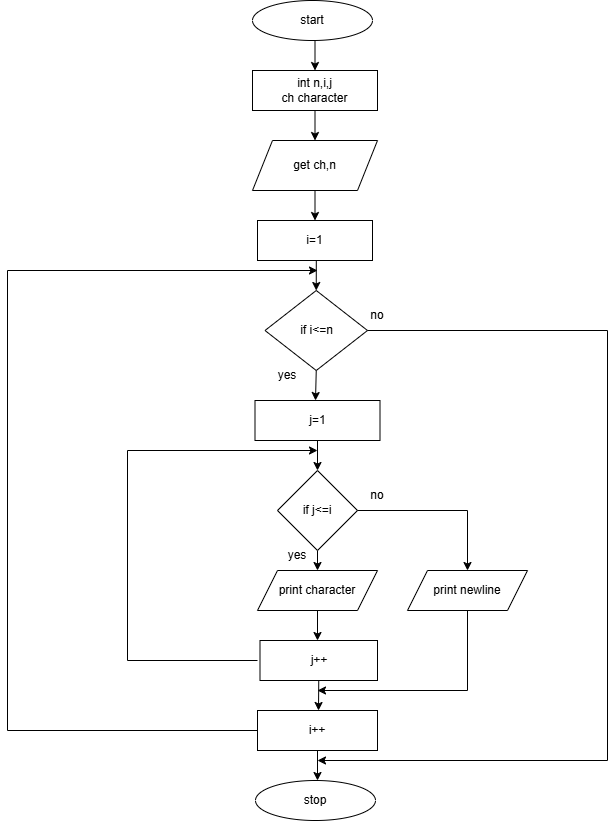
Step 10: i++.

Step 11: print ‘\n’

Step 12: if i<n, goto step 6, else continue

Step 13: stop.

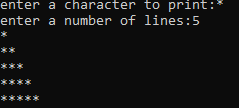
**FLOWCHART:**

****

**SOURCE CODE:**

[**https://github.com/sruthisruthi18112007/c-lab/blob/main/tower.c**](https://github.com/sruthisruthi18112007/c-lab/blob/main/tower.c)

**OUTPUT:**

****

**RESULT:**

Thus the program is compiled and executed successfully with verified output.

|  |  |  |
| --- | --- | --- |
| 8 | **PALINDROME OR NOT** | DATE: 04 Sep 2025 |

**AIM:**

To write a C Program to check if a given positive integer number is a palindrome or not.

**ALGORITHM:**

**Step 1:** Start

**Step 2:** Read a positive integer number (num)

**Step 3:** If num <0,

Display “Please enter a positive integer” and stop

**Step 4:** Set originalNum = num

reversedNum = 0

**Step 5:** If num! =0 Continue, else goto step 10

**Step 6:** remainder = num%10

**Step 7:** reversedNum = (reversedNum\*10) + remainder

**Step 8:** num = num/10

**Step 9:** goto step 5

**Step 10:** If originalNum == reversedNum

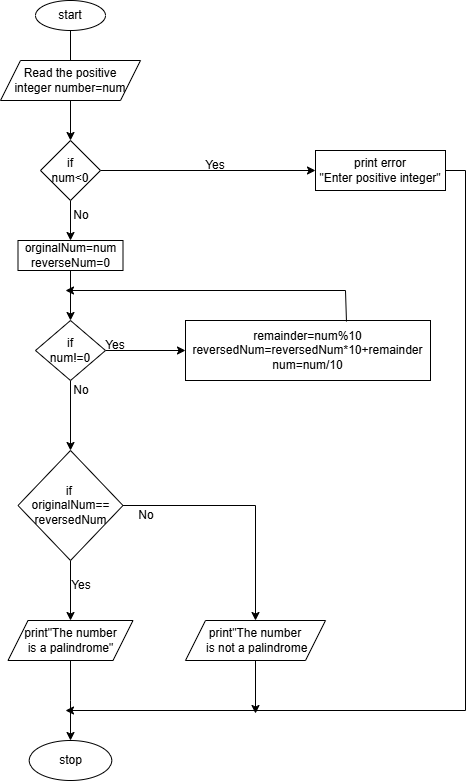
Display “The number is a palindrome” and goto step 11

Else

Display “The number is not a palindrome”

**Step 11:** Stop

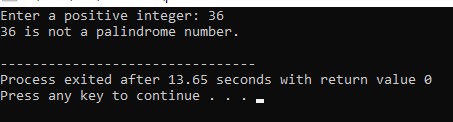
**FLOWCHART:**



**SOURCE CODE:**

[**https://github.com/raghavi2008/c-lab/blob/main/positive%20palindrome%20or%20not.c**](https://github.com/raghavi2008/c-lab/blob/main/positive%20palindrome%20or%20not.c)

**OUTPUT:**

****

**RESULT:**

Thus the program is compiled and executed successfully with verified output

|  |  |  |
| --- | --- | --- |
| 8 | **STRING HANDLING** | DATE:25 Sep 2025 |

**AIM:**

Program to check Implement your own string length and string reversal function

**ALGORITHM:**

**Step 1:** Start

**Step 2:** Read a string (str) from the user

**Step 3:** Print string length (using stringlength () user defined function)

**Step 4:** Print string reverse (using stringreverse () user defined function)

**Step 5:** Stop

**String Length () Algorithm:**

**Step 1:** length=0

**Step 2:** If str [length] is not equal to the null charcter (‘\0’)

length=length+1, and repeat step 2, else goto step 3

**Step 3 :** Return length

**String Reverse () Algorithm:**

**Step 1:** Initialize start=0 and end=len-1

**Step 2:** Repeat if start<end, else goto step 9

**Step 3:** store str[start] in temp

**Step 4:** str[start]=str[end]

**Step 5:** str[end]=temp

**Step 6:** start=start+1

**Step 7:** end=end-1

**Step 8:** Goto step 2

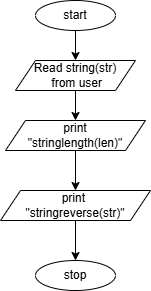
**Step 9:** Output

Display the original string length

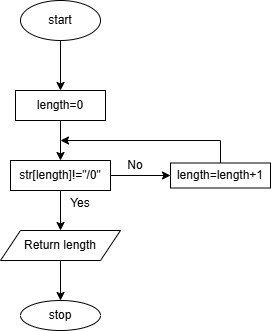
Display the reversed string

**Step 10:** Stop

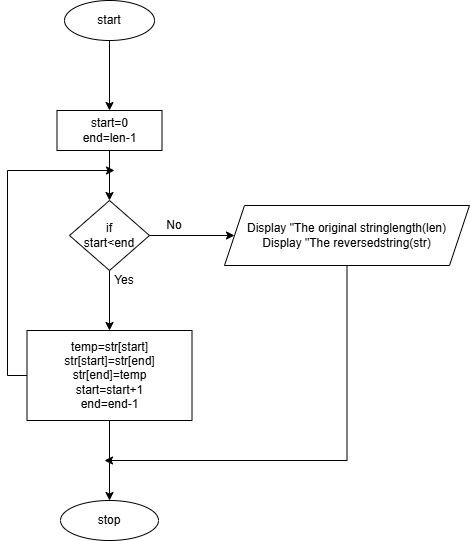
**FLOWCHART:**



**Flowchart for Stringlength(str):**



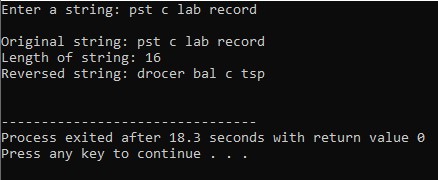
**Flowchart for StringReverse (str, len):**

****

**SOURCE CODE:**

<https://github.com/raghavi2008/c-lab/blob/main/string%20handling.c>

**OUTPUT:**



**RESULT:**

Thus the program is compiled and executed successfully with verified output

|  |  |  |
| --- | --- | --- |
| 9 | **ARRAY SORTING** | DATE: 25 Sep 2025 |

**AIM:**

To write in C Program to sort an array.

**ALGORITHM:**

**Step 1:** Declare counter variables (i, j) temporary variable (temp), array to store elements.

**Step 2:** Read number ofelement n.

**Step 3:** If (n<=0 || n>100) goto step 21 else continue.

**Step 4:** i=0

**Step 5:** Read arr(i)

**Step 6:** If i<n goto step 5 else continue.

**Step 7:** i=0

**Step 8:** j=0

**Step 9:** If arr[j]>arr[j+1] continue else goto step 13.

**Step 10:** temp=arr[j]

**Step 11:** arr[j]=arr[j+1]

**Step 12:** arr[j+1] =temp

**Step 13:** j=j+1

**Step 14:** If j<(n-1-i) goto step 9 else continue

**Step 15:** i=i+1

**Step 16:** If i<n-1 goto step 8 else continue

**Step 17:** i=0

**Step 18:** Printf arr(i)

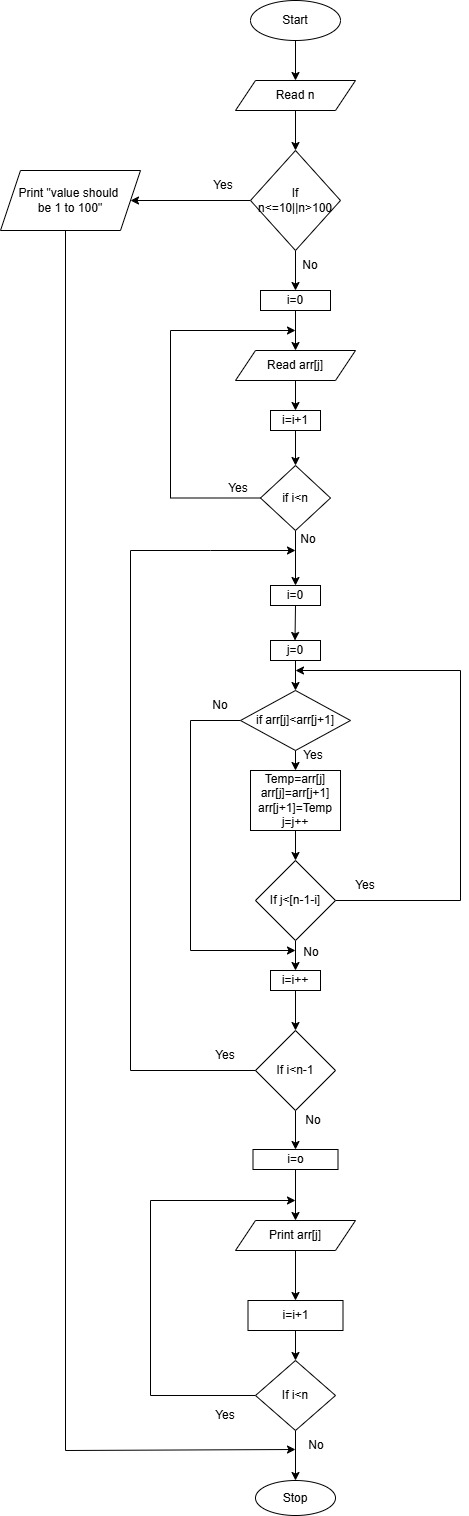
**Step 19:** i=i+1

**Step 20:** If i<n goto step 18, else goto step 22

**Step 21:** Print “n value should be 1 to 100”

**Step 22:** Stop

**FLOWCHART:**

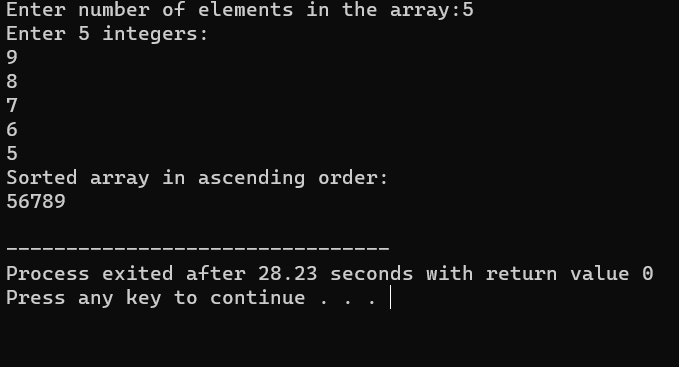


# 

# **SOURCE CODE:**

[c-lab/arraysorted.c at main · rajlaxmi8327/c-lab](https://github.com/rajlaxmi8327/c-lab/blob/main/arraysorted.c)

**OUTPUT:**

****

**RESULT :**

Thus the program is compiled and executed successfully with verified output.

|  |  |  |
| --- | --- | --- |
| 10 | **FACTORIAL** | DATE:09 Oct 2025 |

**AIM:**

Write a C program to find the factorial of an integer using recursive function

**ALGORITHM:**

**Step 1:** Start

**Step 2:** Read limit, n

**Step 3:** If n< 0 print “Give positive numbers” and go to step 5 else continue

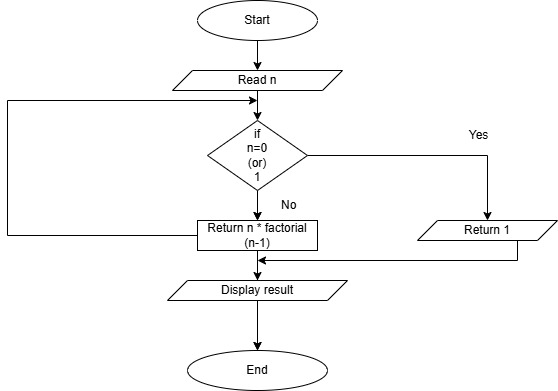
**Step 4:** Print factorial (n)

**Step 5:** Stop

Algorithm for factorial (int n)

1. If n=0 or n=1, return 1else, return n\*factorial (n-1)

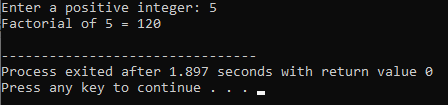
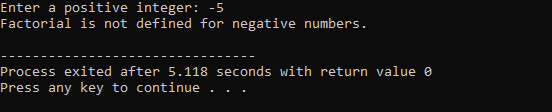
**FLOWCHART:**



**SOURCE CODE:**

<https://github.com/vaibav77/c-lab-vaibav/blob/main/factorial.c>

**OUTPUT:**

** **

**RESULT:**

Thus the program is compiled and executed successfully with verified output.

|  |  |  |
| --- | --- | --- |
| 11 | **SWAPPING OF NUMBERS** | DATE: 09 Oct 2025 |

**AIM:**

Program to swap two values using function pointer

**ALGORITHM:**

**Step 1:** Start

**Step 2:** Declare variable int x, y, temp

**Step 3:** Get x, y

**Step 4:** x = temp

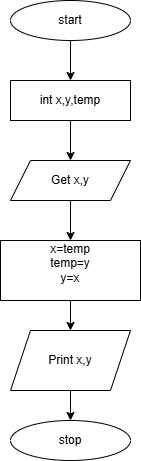
**Step 5:** y = x

**Step 6:** temp = y

**Step 7:** Print x, y

**Step 8:** Stop

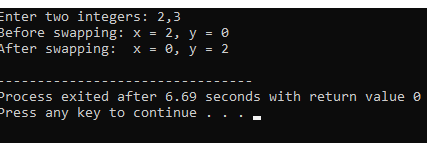
**FLOWCHART:**

****

**SOURCE CODE:**

[**https://github.com/112534037/c-lab/blob/main/swapping%20of%20numbers.c**](https://github.com/112534037/c-lab/blob/main/swapping%20of%20numbers.c)

**OUTPUT:**

****

**RESULT:**

Thus the program is compiled and executed successfully with verified output.

|  |  |  |
| --- | --- | --- |
| 12 | **USING STRUCTURES** | DATE: 23 Oct 2025 |

**AIM :**

To Program to store student’s information using structure

**ALGORITHM:**

**Step 1:** Start the program

**Step 2**: Define a structure named Student with the following members:

roll No → integer

name → character array (string)

marks → float

**Step 3**: Declare a variable s of type struct Student.

**Step 4**: Input the student’s roll number and store it in s. rollno.

**Step 5**: Input the student’s name and store it in s.name.

**Step 6**: Input the student’s marks and store them in s. marks.

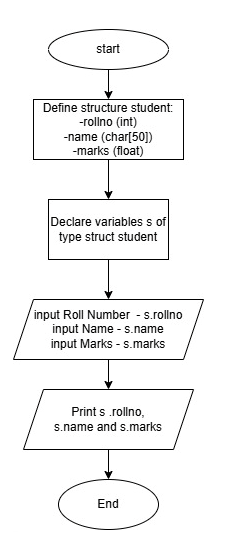
**Step 7**: Display the message “--- Student Information ---”.

**Step 8**: Print the stored values:

Roll Number, Name, Marks.

**Step 9**: End the program.

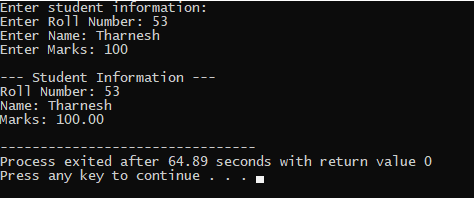
**FLOWCHART:**



**SOURCE CODE:**

[**https://github.com/Tharnesh777/c-lab/blob/main/STRUCTURE.c**](https://github.com/Tharnesh777/c-lab/blob/main/STRUCTURE.c)

**OUTPUT:**

****

**RESULT :**

Thus the program is compiled and executed successfully with verified output.

|  |  |  |
| --- | --- | --- |
| 13 | **FILE PROGRAMMING** | DATE:23 Oct 2025 |

**AIM:**

To Write a C Program to perform read and write operation on a file.

**ALGORITHM:**

**Step 1:** Start the Program.

**Step 2**: Declare filename (filename), character (ch),

**Step 3**: Read the filename

**Step 4**: Open file in Write mode using fopen (filename,” w”)

**Step 5**: If step 4 Fails, goto step 16 Else continue.

**Step 6:** Read the ch,

**Step 7:** If ch! =’~’ continue else goto step 10.

**Step 8:** Write ch into file using fput(c).

**Step 9:** Repeat the Step 6.

**Step 10:** Close the file.

**Step 11**: Read the Same file in read mode using fopen (filename,” r”)

**Step 12:** If Step 11 to fails goto Step 16, else continue.

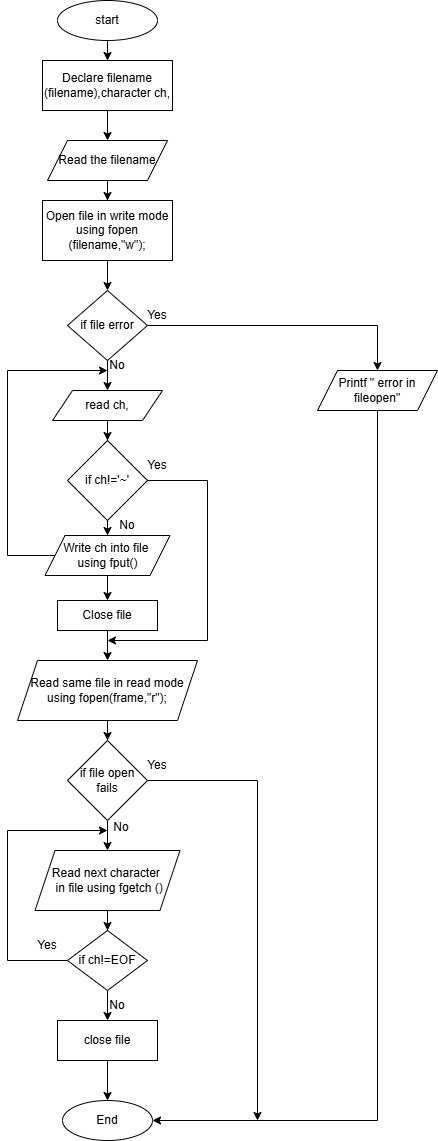
**Step 13**: Read the next character in file using fgetch ()

**Step 14:** if ch! =EOF repeat Step 13 else goto Step 15.

**Step 15:** Close the file.

**Step 16:** End the Program.

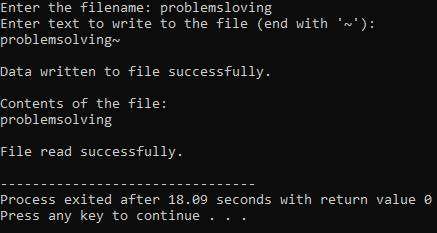
**FLOW CHART:**



**SOURCE CODE:**

<https://github.com/Tharnesh777/c-lab/blob/main/READ%20AND%20WRITE.c>

**OUTPUT:**

****

**RESULT:**

Thus the program is compiled and executed successfully with verified output.