



# FLOWGORITHM LAB SESSION #F3

CTPS 2018

# STRING IS AN ARRAY OF CHARACTERS

## String Functions in FLOWGORITHM

### String length

Len(s)

Returns length  
of a string

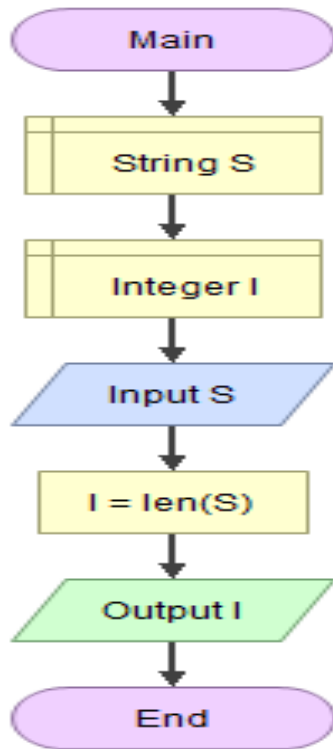
### Character extraction

Char(s, i)

Returns a character from  
the string s at index i.

Characters are indexed  
starting at 0.

# STRING LENGTH



Declare String S

Declare Integer l

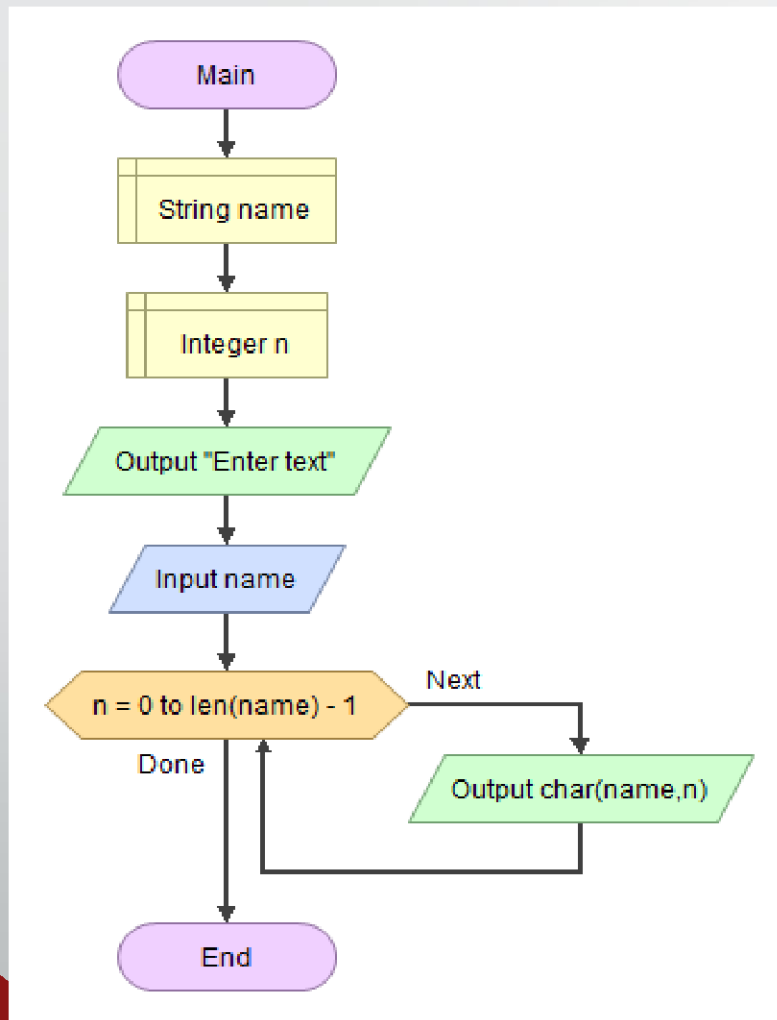
Input S

Set  $l = \text{len}(S)$

Display l

alliswell

9



String to letters

# EXTRACT CHARACTERS

Declare String S

Declare String C

Declare Integer L

Declare Integer I

Input S

Set C = ""

Set L = len(S)

For I = 0 To L - 1

Set C = substring(S, I, 1)

Display C

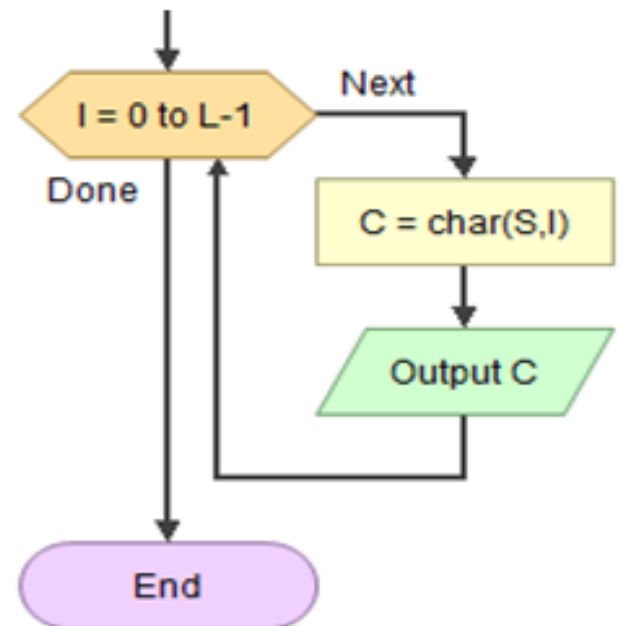
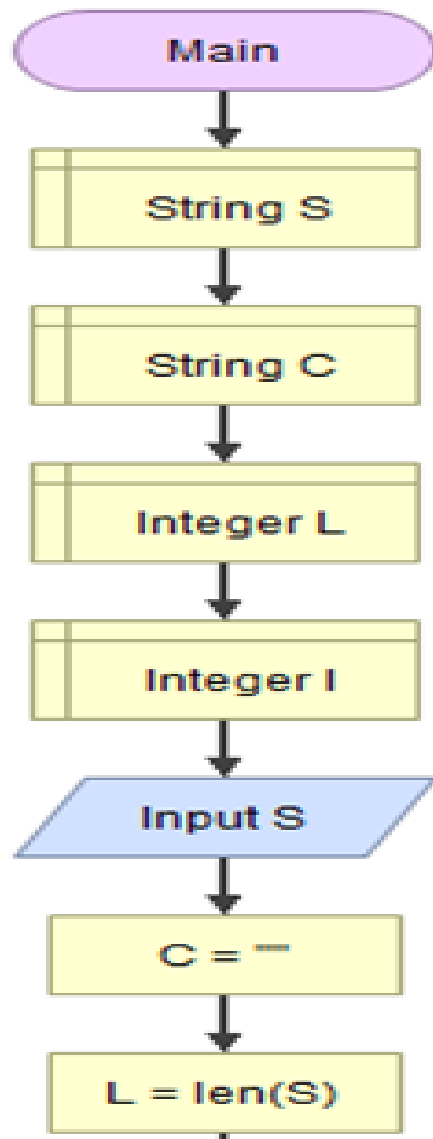
End For

dfg

d

f

g



# STRING COPY

Declare String S

Declare String SS

Declare Integer L

Declare Integer I

Input S

Set SS = ""

Set L = len(S)

For I = 0 To L - 1

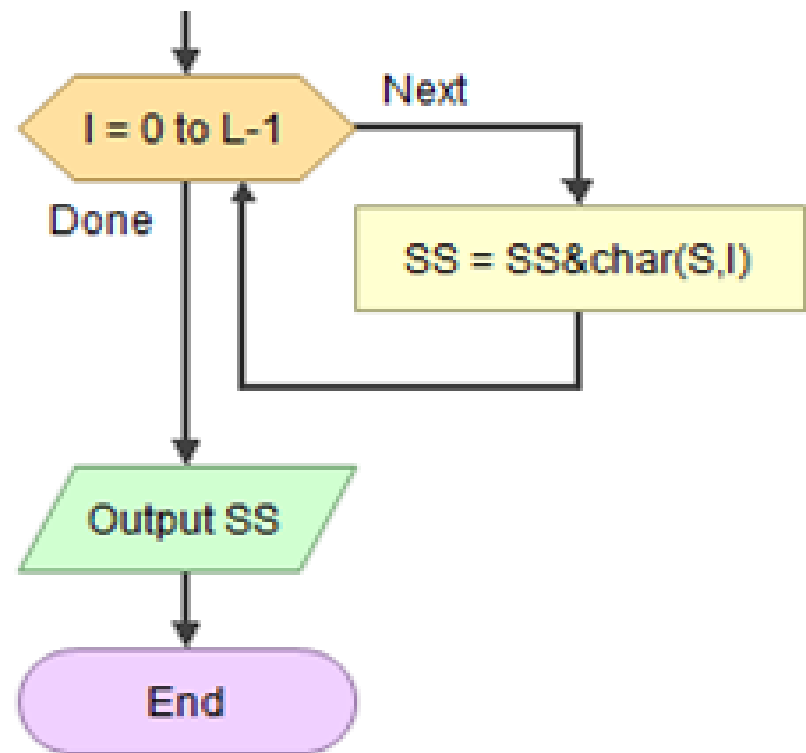
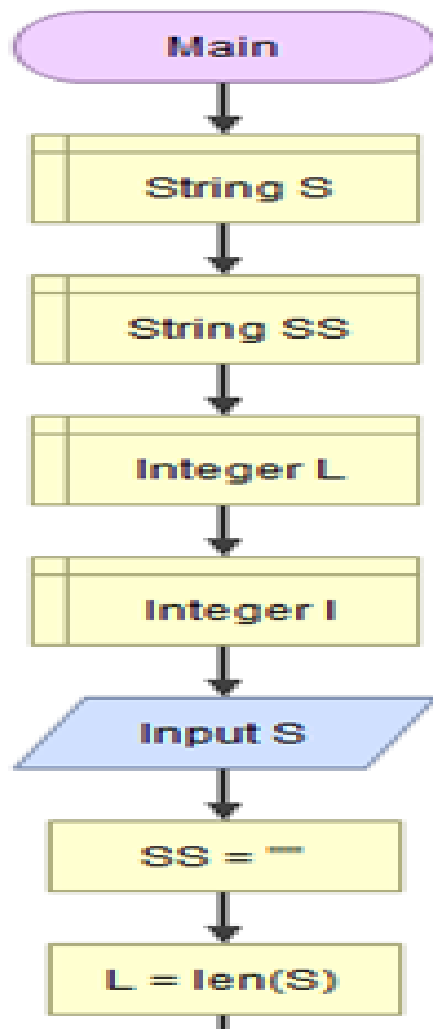
Set SS = SS + substring(S, I, 1)

End For

Display SS

SELL

SELL





# STRING REVERSE

Declare String S

Declare Integer l

Declare Integer i

Declare String Rev

hello

Input S

Set Rev = ""

Set l = len(S)

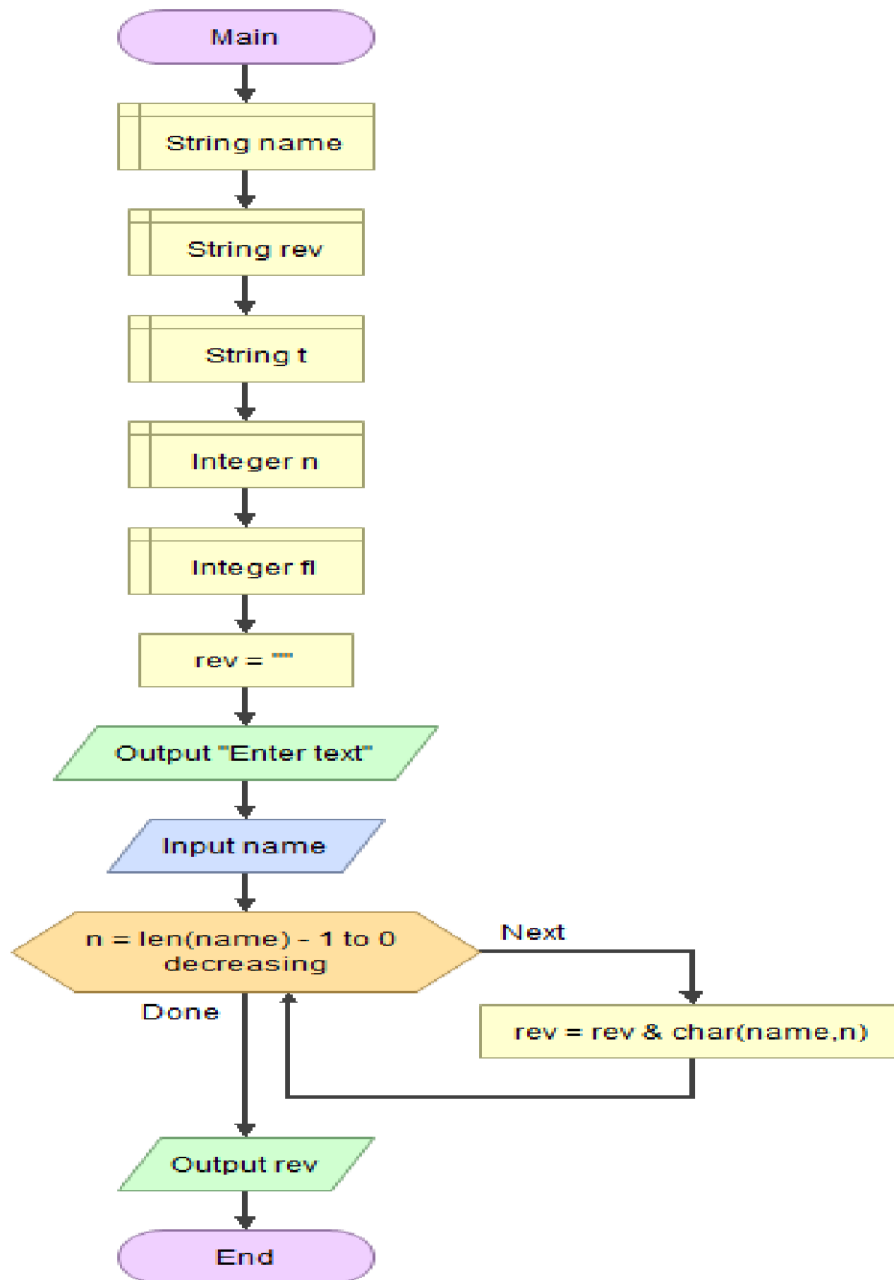
For i = l - 1 To 0 Step -1

Set Rev = Rev + substring(S, i, l)

End For

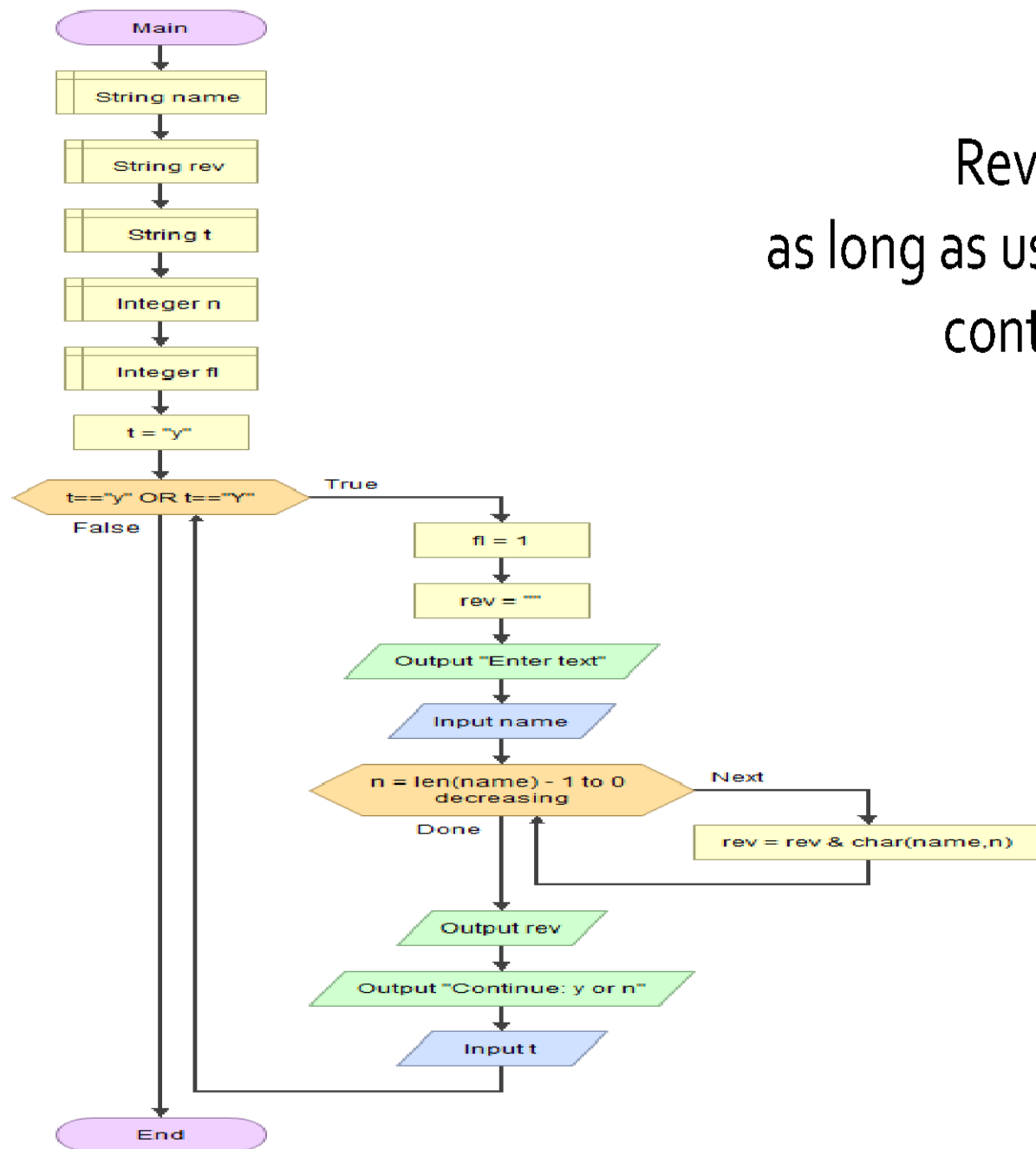
Display Rev

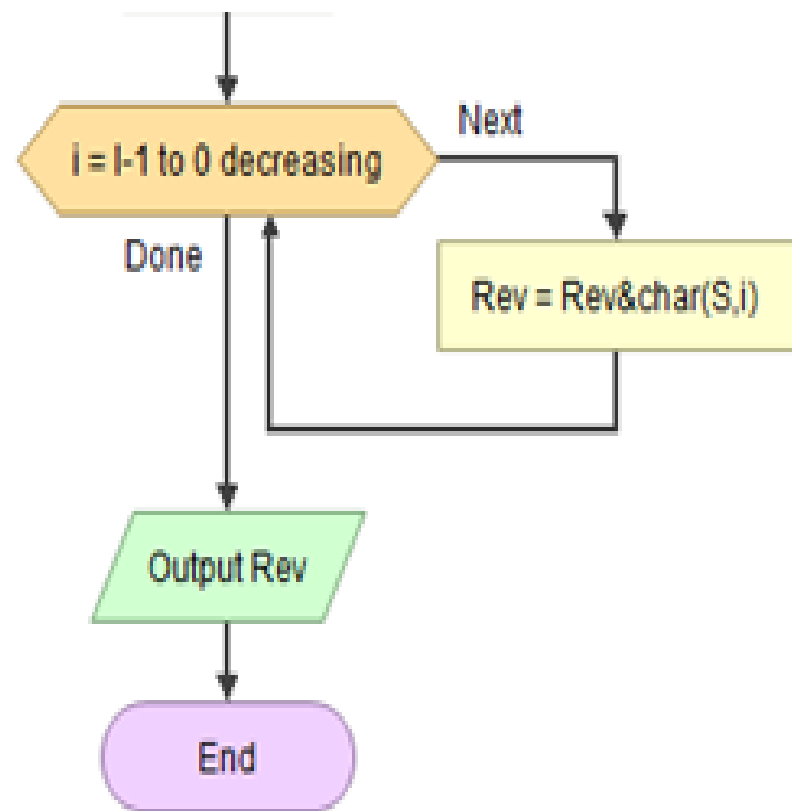
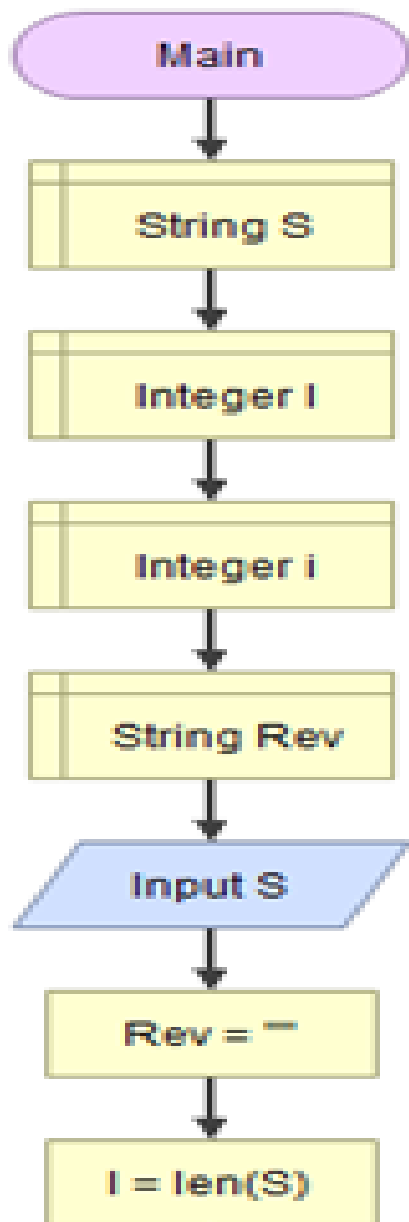
olleh



Reversing a string

Reversing strings  
as long as user wishes to  
continue





# PALINDROME

Module main()

Declare String S

Declare Integer l

Declare Integer i

Declare String RS

Input S

Set RS = ""

Set RS = Reverse(S)

If RS == S Then

Display "PALINDROME"

Else

Display "NOT PALINDROME"

End If

End Module

Function String Reverse(String SS)

Declare String Rev

Declare Integer l

Declare Integer i

Set l = len(SS)

Set Rev = ""

For i = l - 1 To 0 Step -1

Set Rev = Rev + substring(SS, i, l)

End For

Return Rev

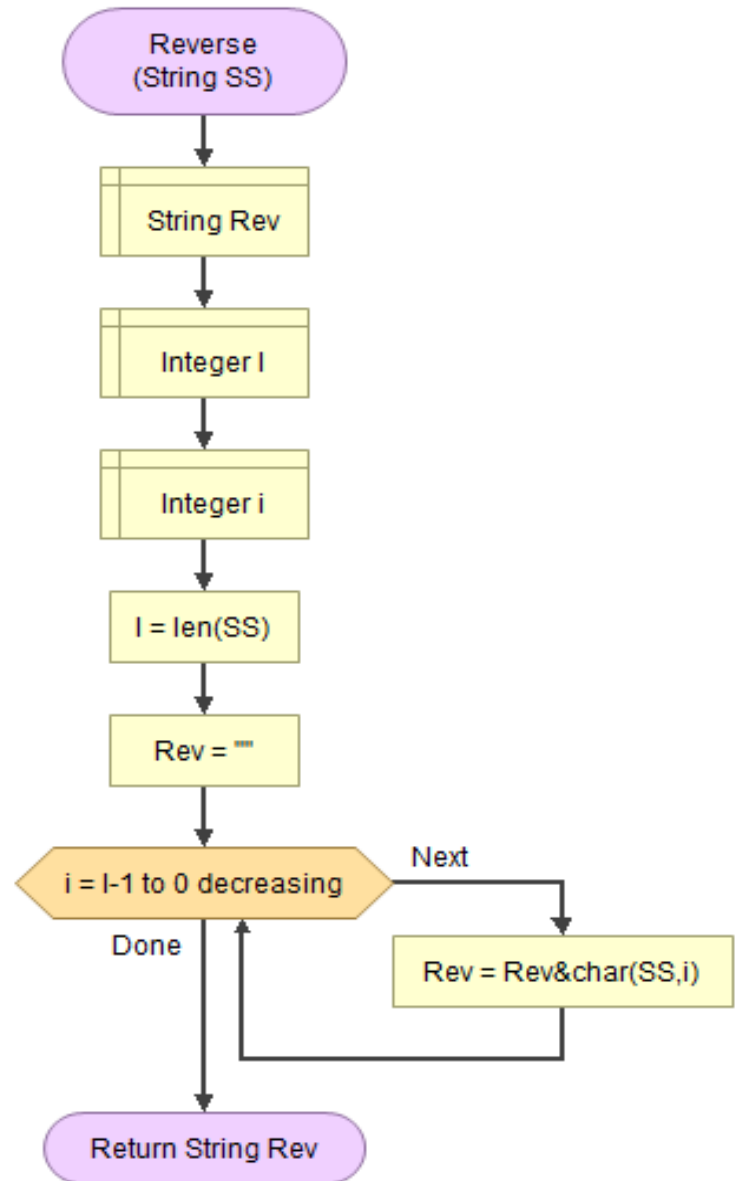
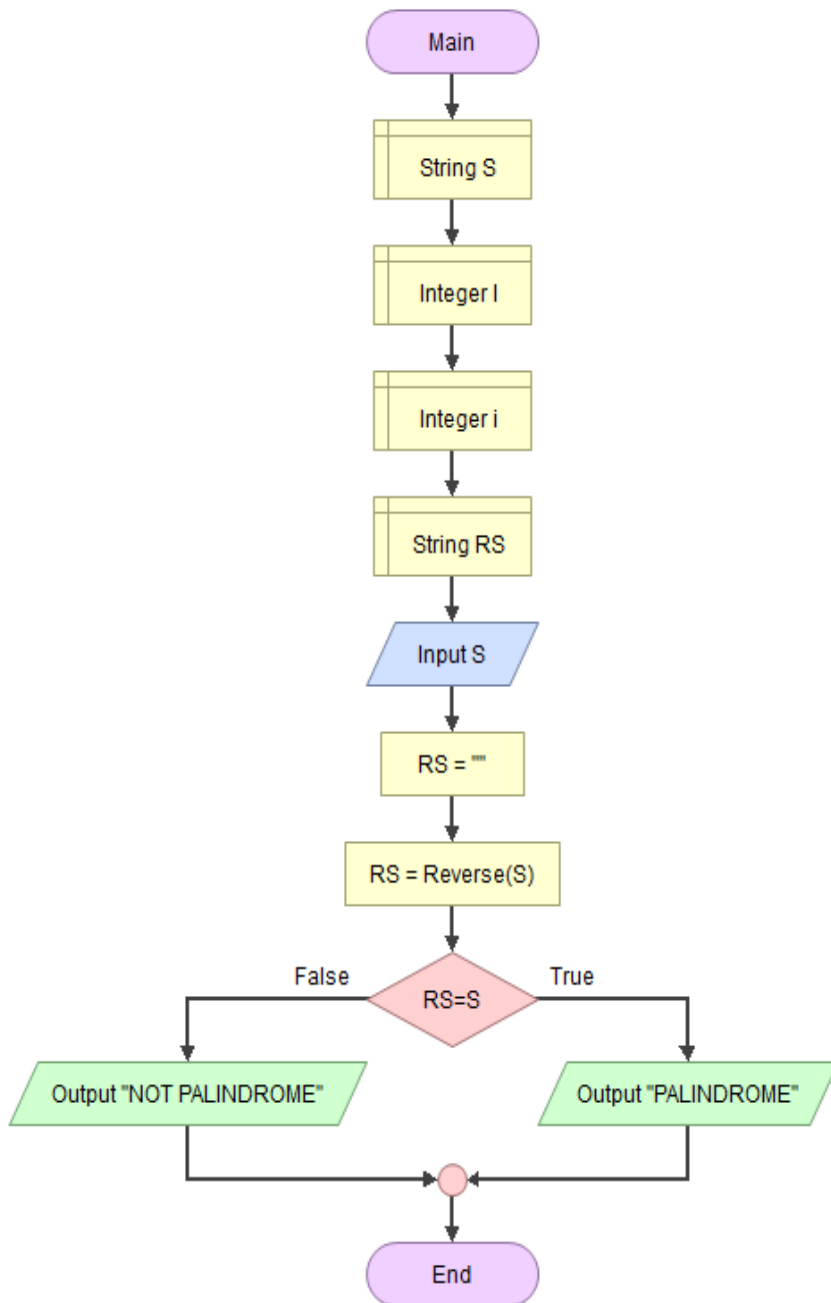
End Function

ammas

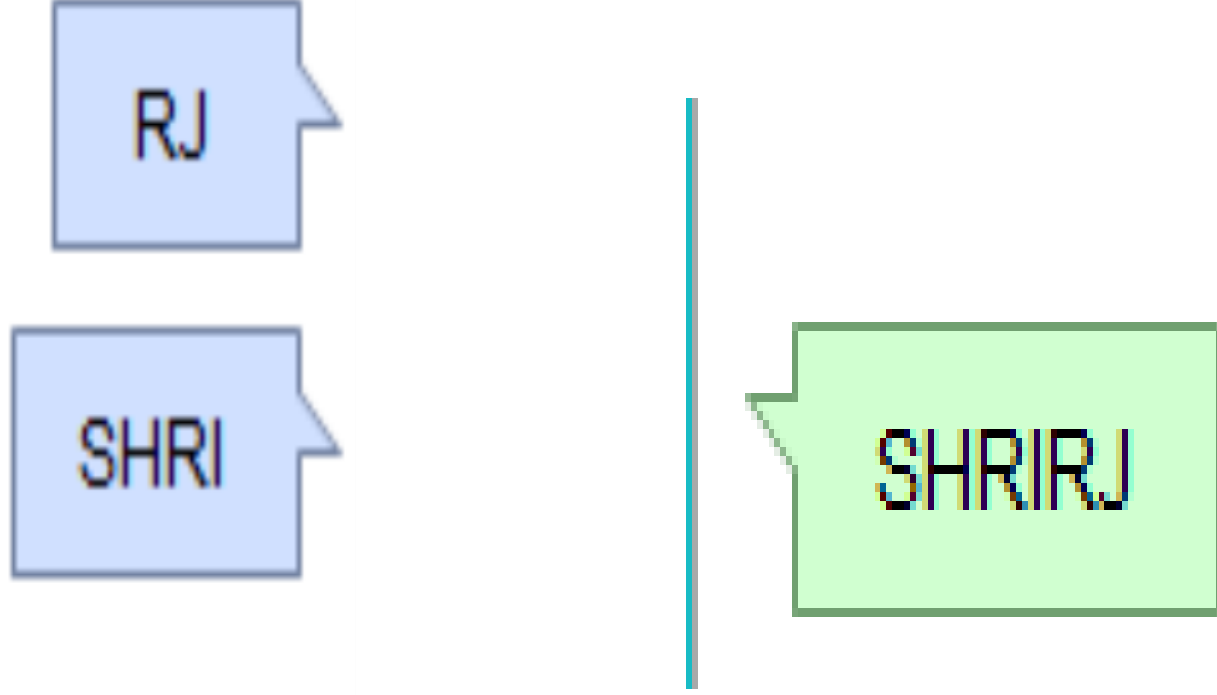
NOT PALINDROME

MALAYALAM

PALINDROME



# STRING CONCATENATE





Declare String S

Declare String SS

Declare Integer LS

Declare Integer LSS

Declare Integer I

Declare String C

Input S

Input SS

Set LS = len(S)

Set LSS = len(SS)

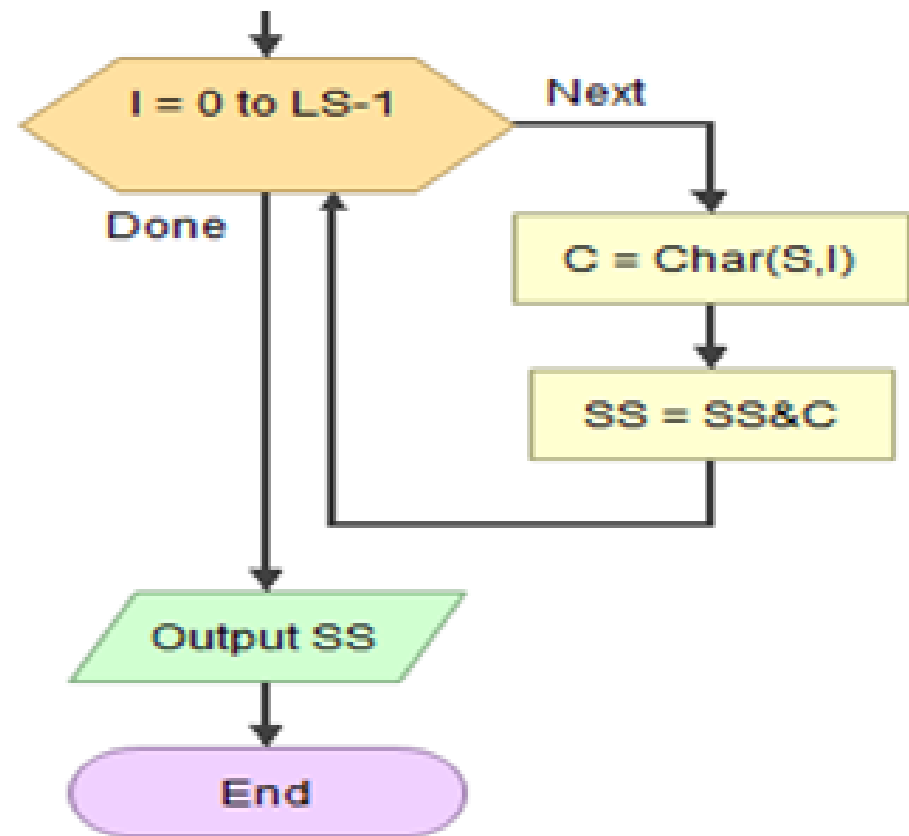
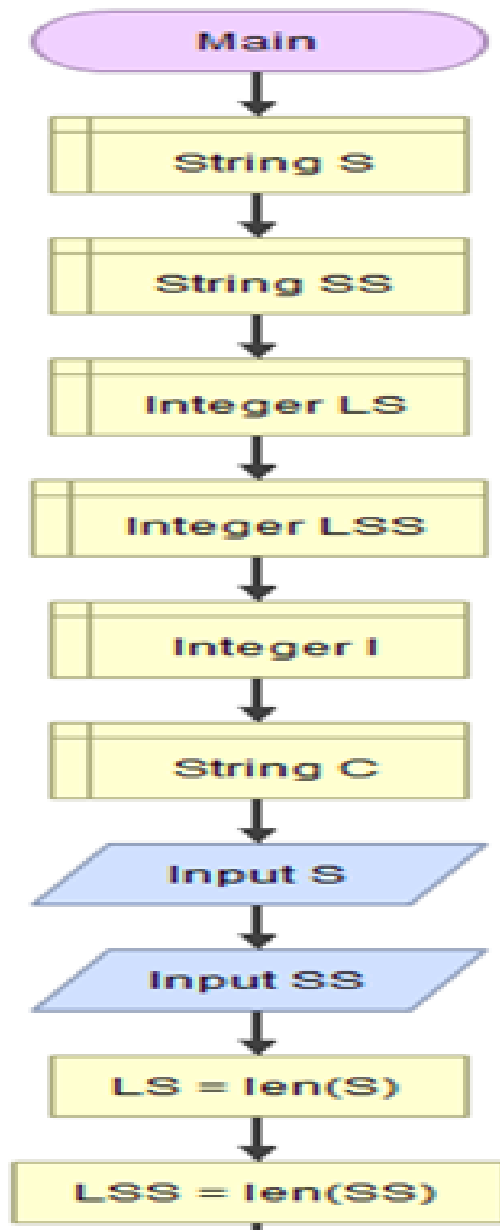
For I = 0 To LS - 1

Set C = substring(S, I, 1)

Set SS = SS + C

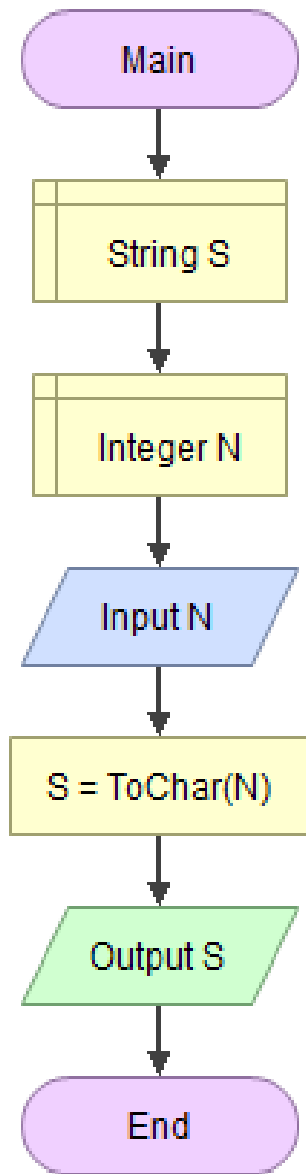
End For

Display SS



ToChar(n) Convert a character code n into an character

ToCode(c) Convert a character c into a character code (integer).



---

Declare String S

Declare Integer N

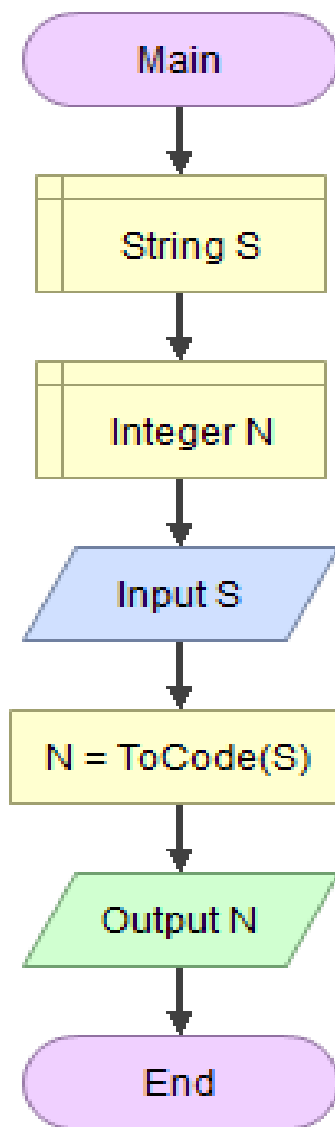
Input N

Set S = toChar(N)

Display S

65

A



Declare String S

Declare Integer N

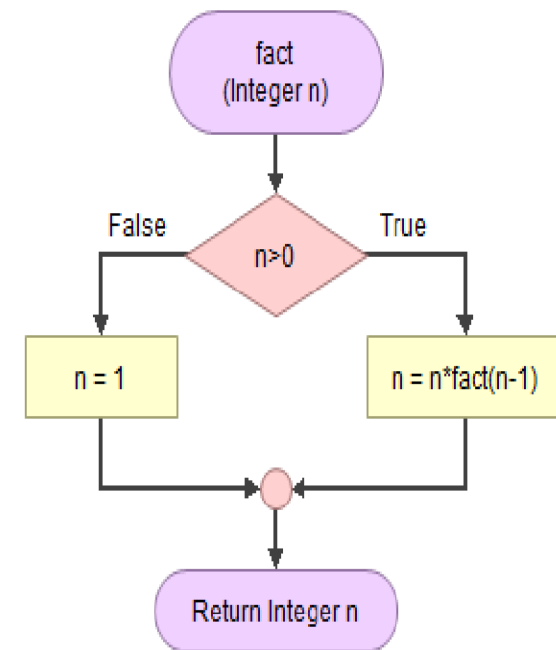
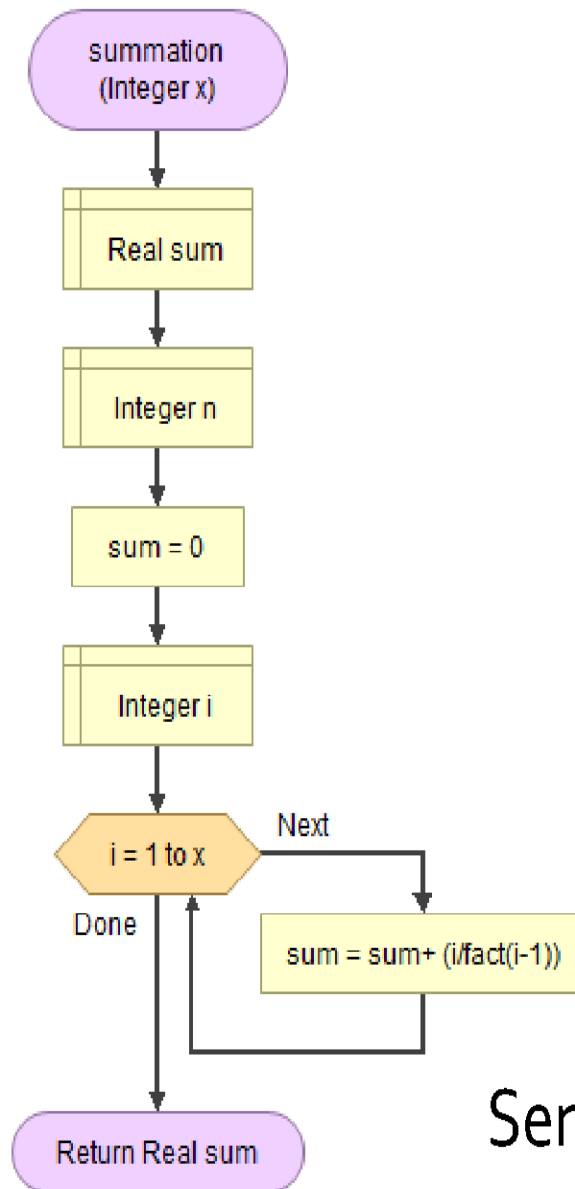
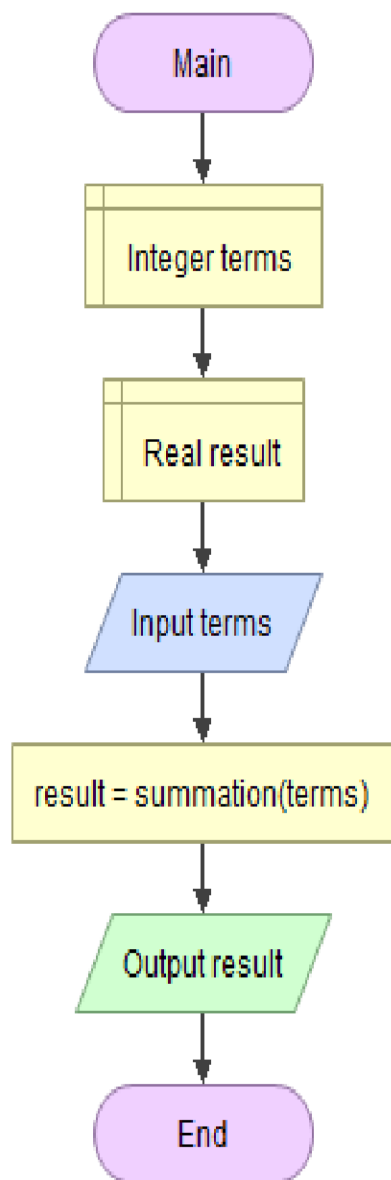
Input S

Set N = toCode(S)

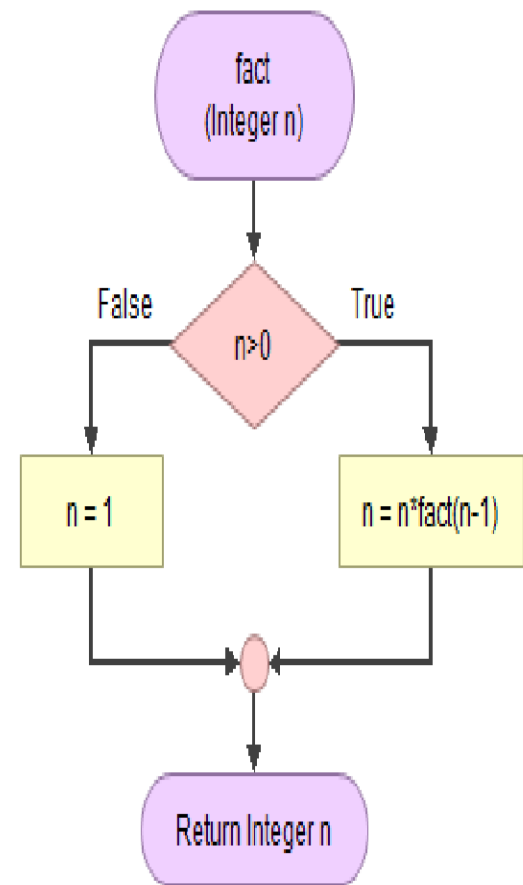
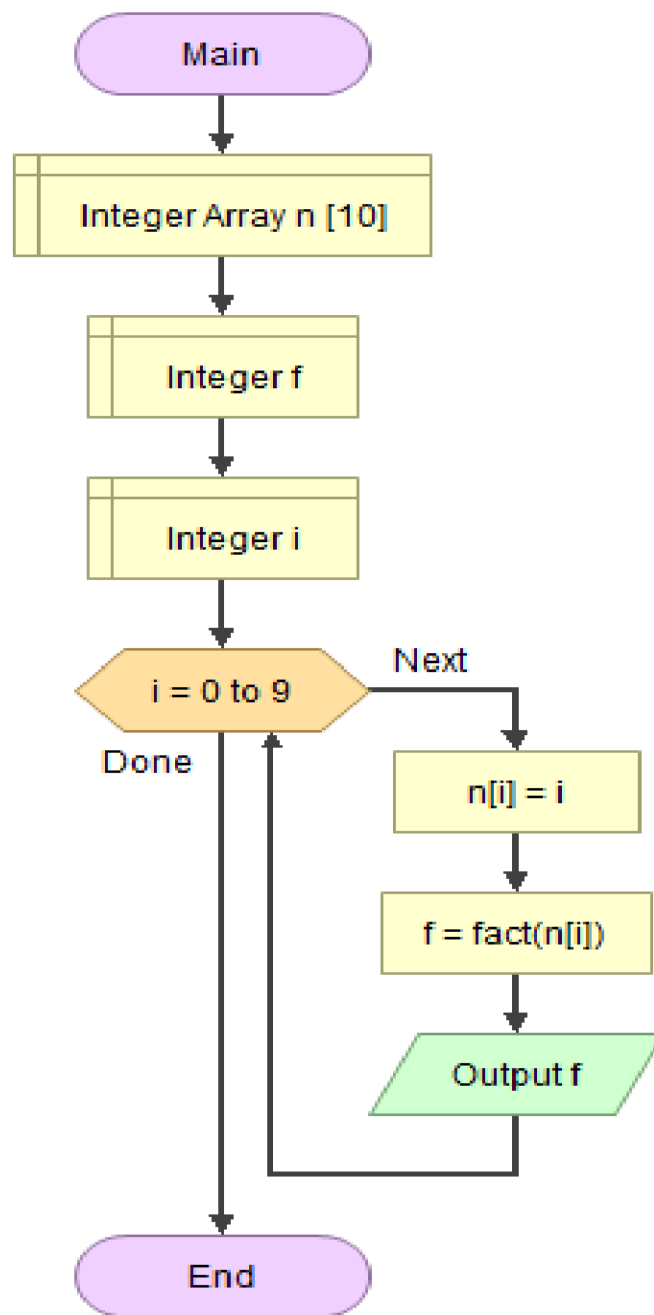
Display N

a

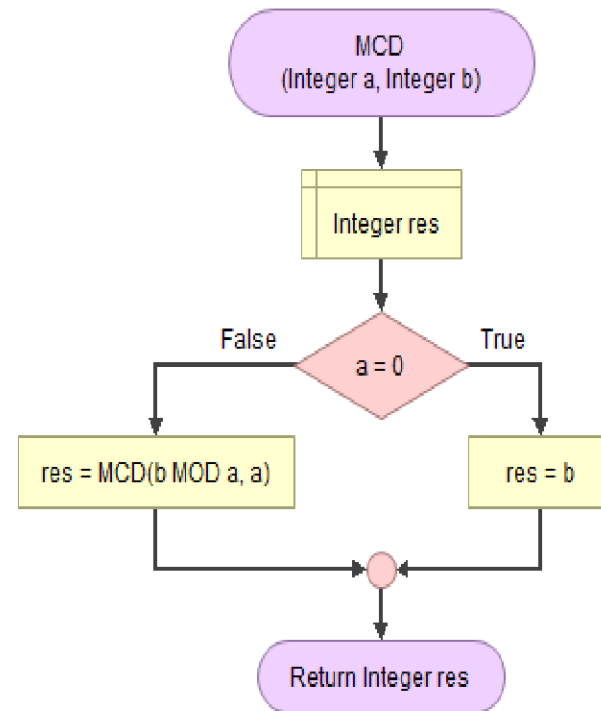
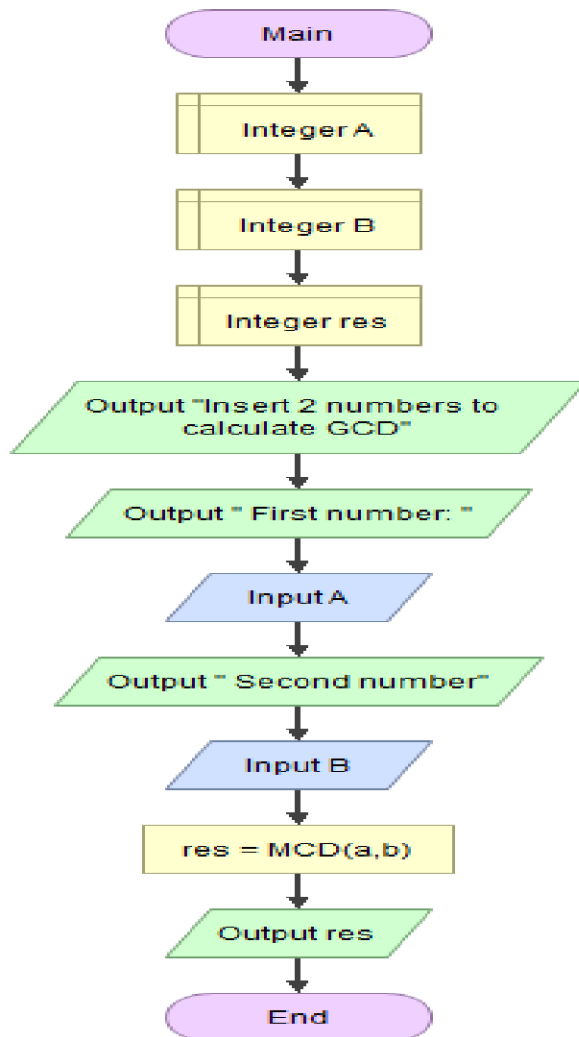
97



Series summation using  
Recursion

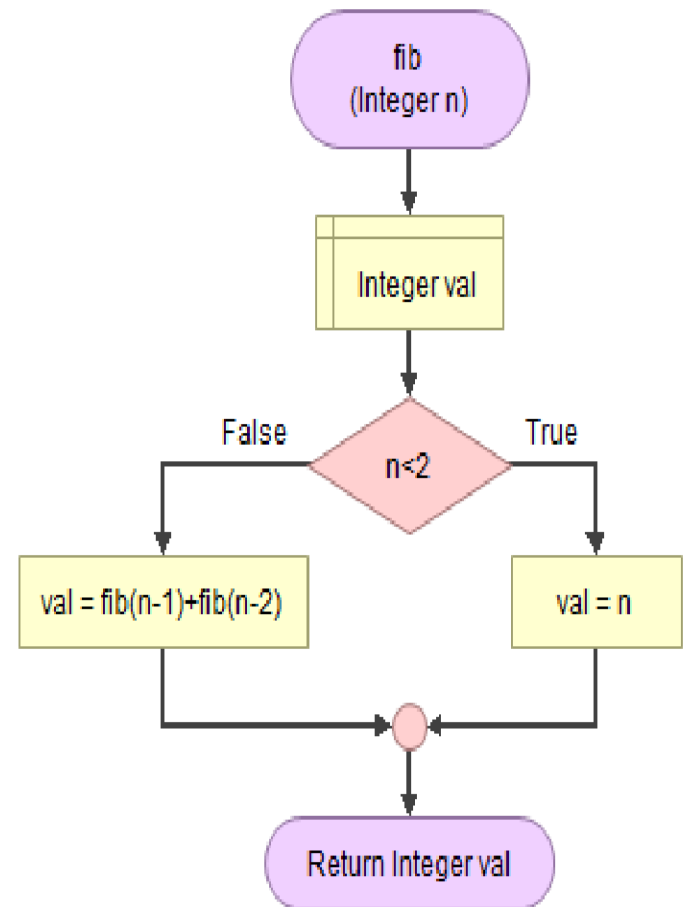
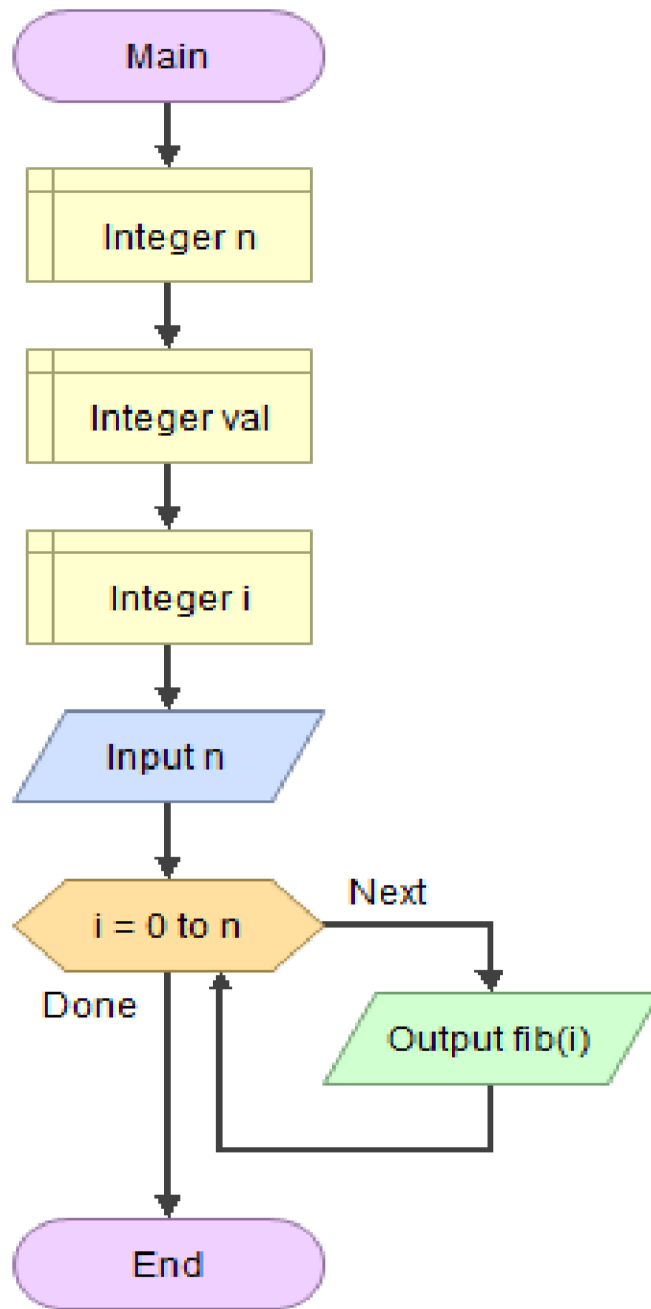


Factorial using Recursion

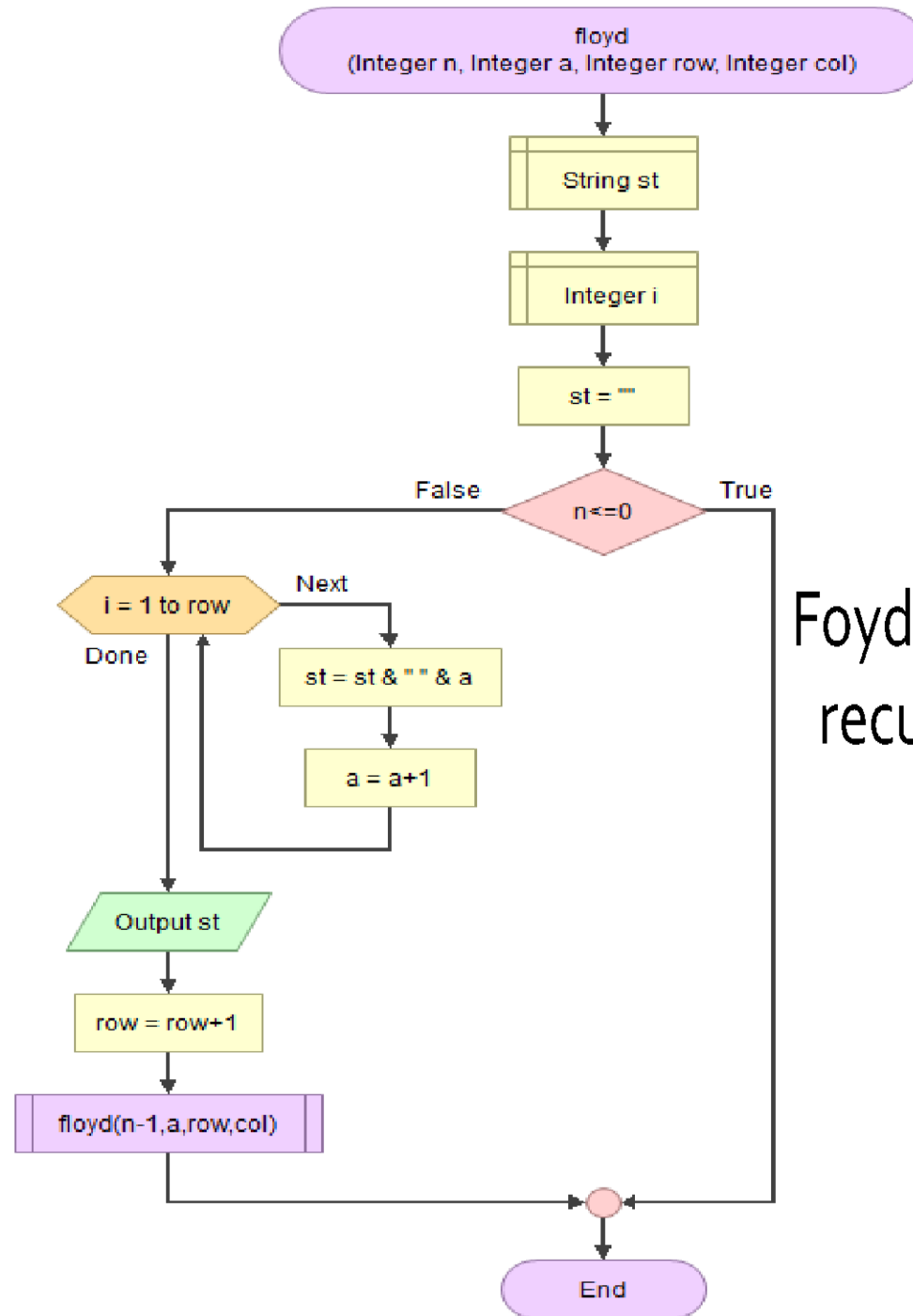
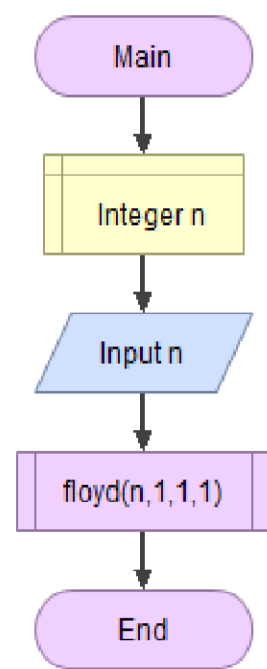


GCD using recursion

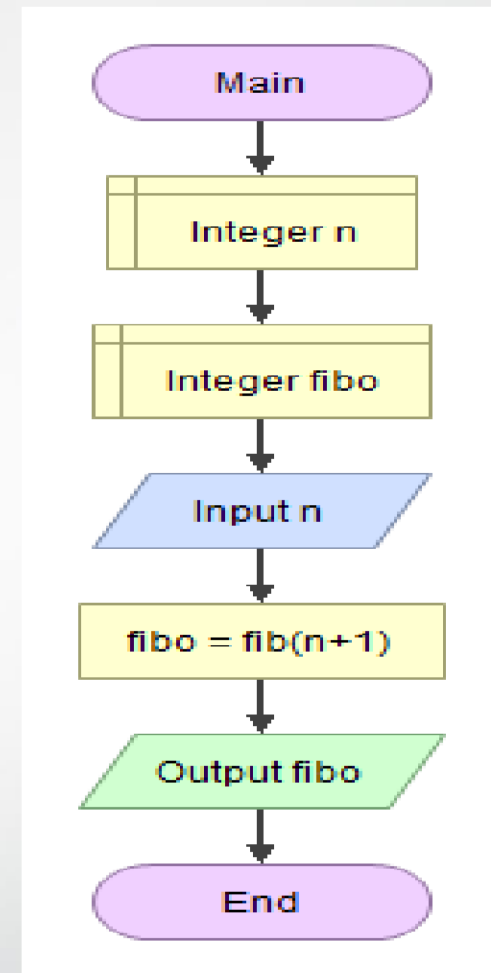
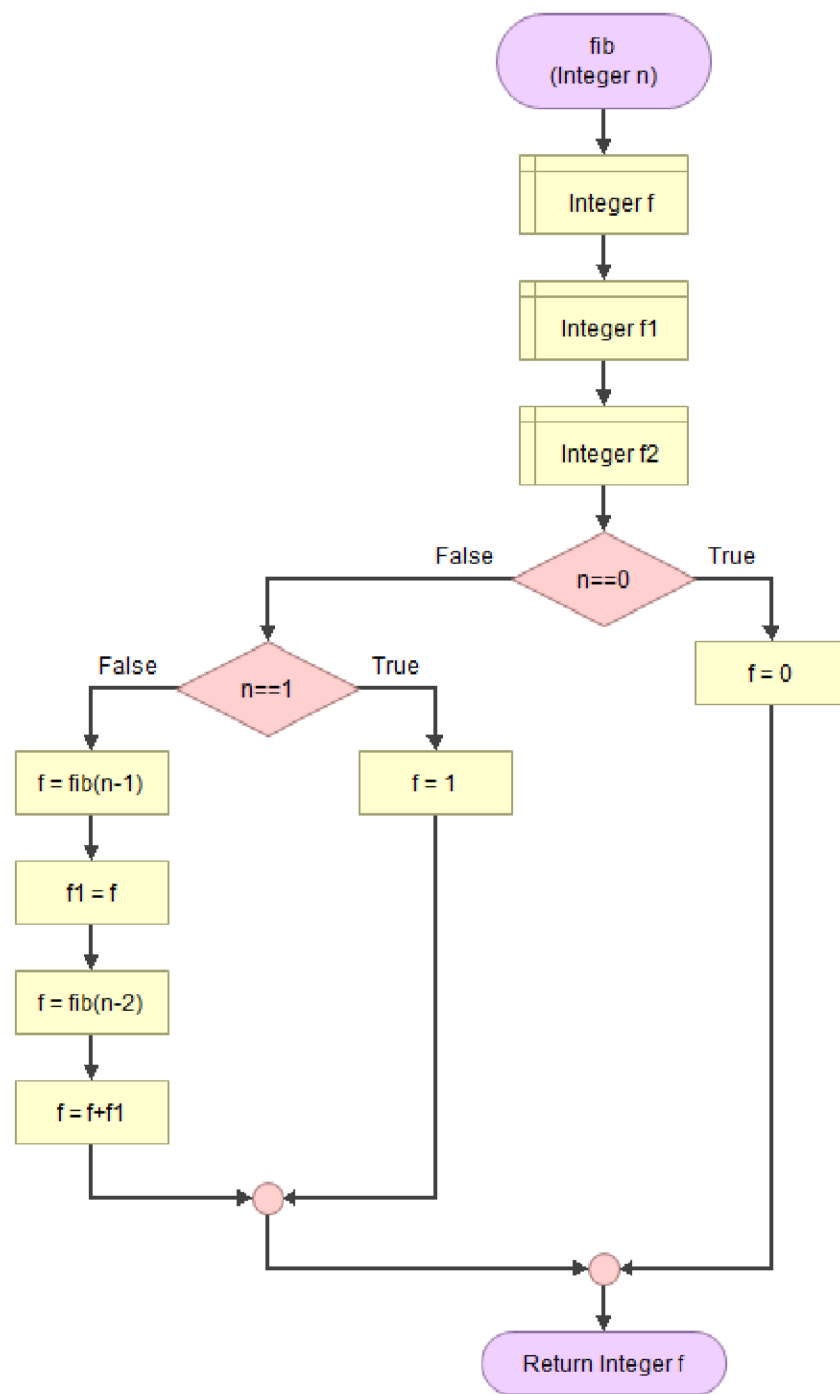




Fibonacci using Recursion



Foyd's Triangle using recursion



Fibonacci: nth term using  
Recursion



#F3

Take Home Exercise

## Question 1

Print array elements using recursion



## Question 2

Summation of first n numbers using recursion



## Question 3

Tower of Hanoi using recursion



## Question 4

Find and return the minimum element in an array using recursion



## Question 5

The Hailstone sequence is created using the following rules:

- Start with any positive whole number, called the seed.
- If the number is even, halve it ( $\frac{1}{2}n$ ) to get the next number in the sequence.
- If the number is odd, multiply it by 3 and then add 1 ( $3n + 1$ ) to get the next number in the sequence.
- Continue until you see repetition.

For  $n=5$ , we have **5, 16, 8, 4, 2, 1, 4, 2, 1, ...**

## Generate the Hailstone sequence using recursion

The Hailstone sequence of numbers can be generated from a starting positive integer,  $n$  by:

If  $n$  is 1 then the sequence ends.

If  $n$  is even then the next  $n$  of the sequence  $= n/2$

If  $n$  is odd then the next  $n$  of the sequence  $= (3 * n) + 1$