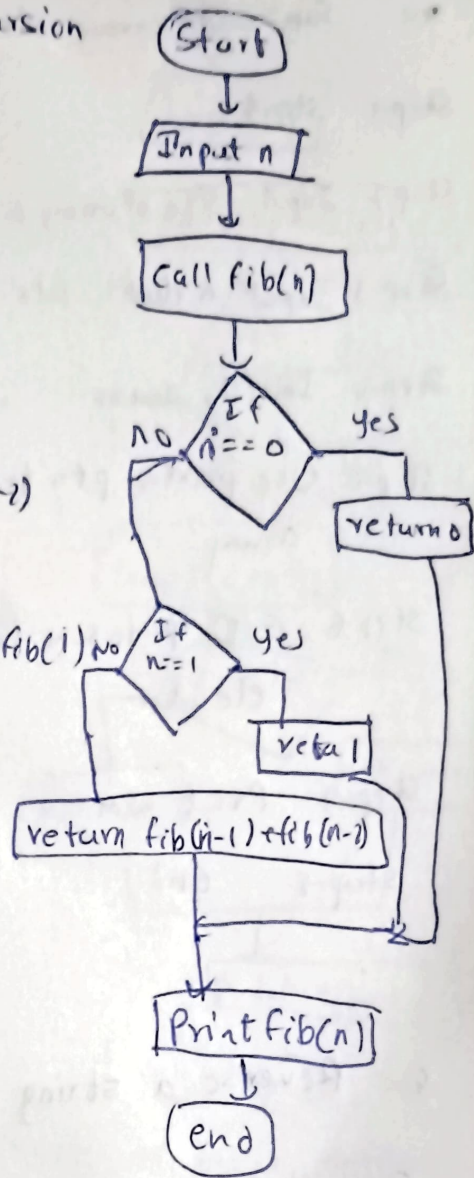


### 47. Fibonacci series using recursion

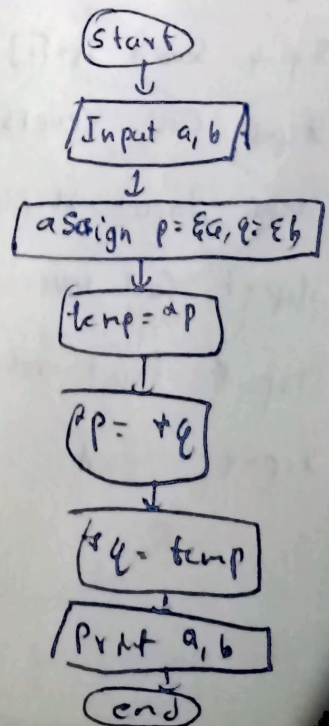
Algorithm

- Step-1 Start
- Step-2 Define function fib(n)
- Step-3 If  $n == 0 \rightarrow$  return 0
- Step-4 If  $n == 1 \rightarrow$  return 1
- Step-5 else return  $\text{fib}(n-1) + \text{fib}(n-2)$
- Step-6 Input number of terms n
- Step-7 loop from 0 to n-1 print fib(i)
- Step-8 end



### 43. Swap two numbers using pointers.

- Step-1 Start
- Step-2 Input two numbers a and b.
- Step-3 use pointers \*p and \*q to store address of a and b
- Step-4 Swap using temp.
- Step-5  $\text{temp} = *p$
- Step-6  $*p = *q$
- Step-7  $*q = \text{temp}$
- Step-8 Print swapped no
- Step-9 end



46. Function to check prime

Step 1: Start

Step 2: Define function is prime(n)

Step 3: If  $n \leq 1$  return false

Step 4: for  $i = 2$  to  $n/2$  →

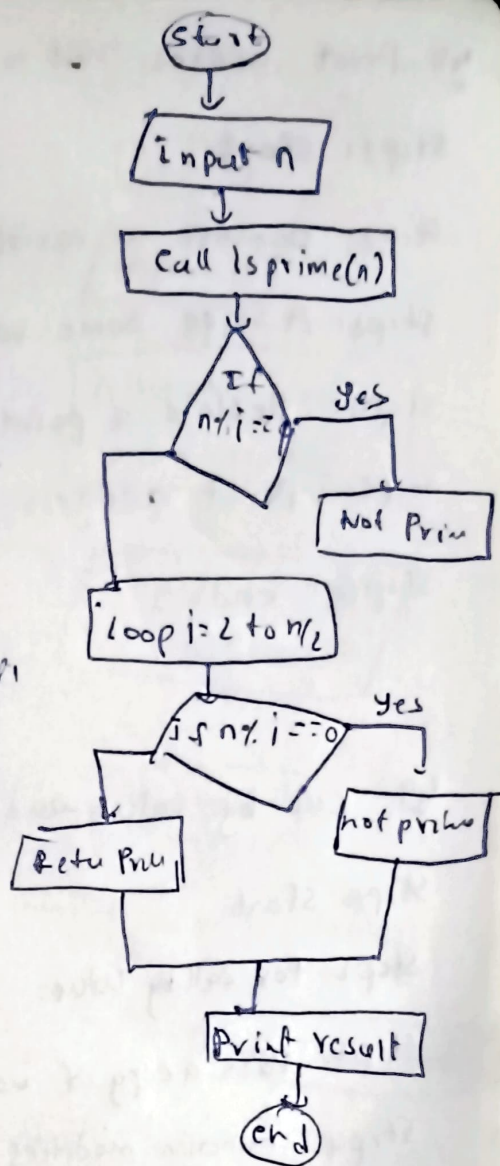
Step 5: if  $n \% i == 0$  → return false

Step 6: else return true.

Step 7: Input n.

Step 8: Call is prime(n) and print result

Step 9: end



47. Function to return maximum of three numbers

Step 1: Start

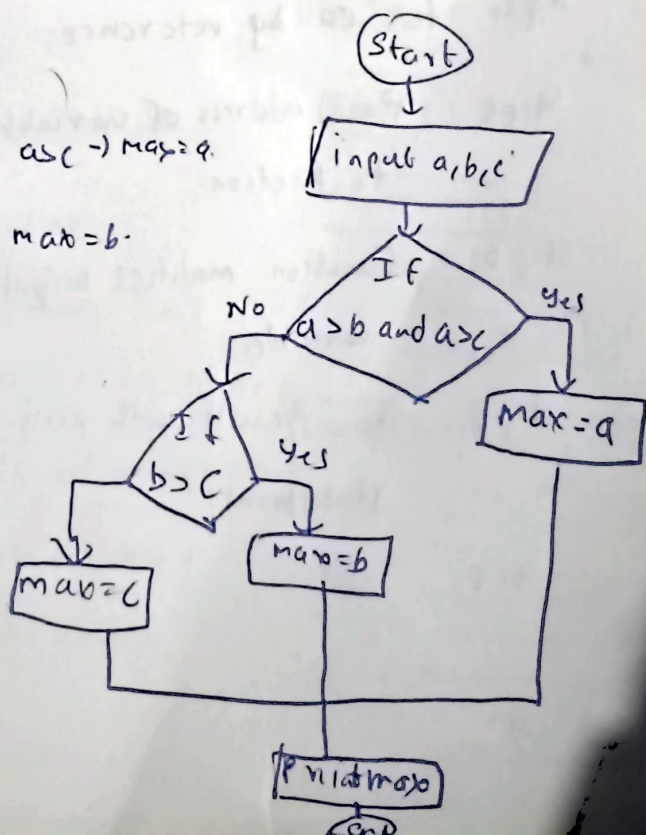
Step 2: Input a, b, c.

Step 3: If  $a > b$  and  $a > c$  →  $max = a$ .

Step 4: Else if  $b > c$  →  $max = b$ .

Step 5: Print max.

Step 6: End.





### 29 Dynamic memory allocation (malloc) and sum of array

Step 1: Start

Step 2: Input Size n

Step 3: Allocate memory dynamically

$arr = \text{malloc}(n * \text{sizeof}(\text{int}))$

Step 4: Input n elements into arr

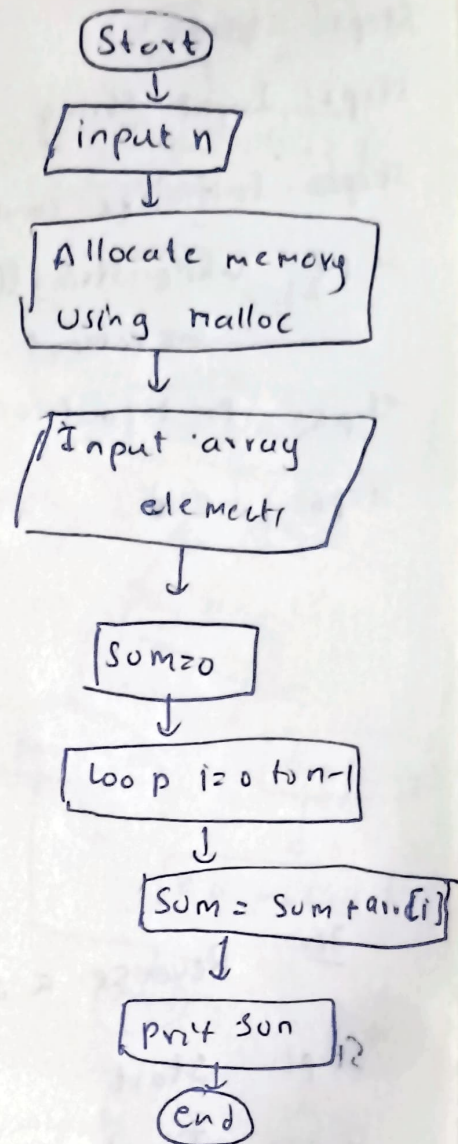
Step 5: initialize sum = 0

Step 6: Loop through array +  
add elements to sum.

Step 7: Print sum

Step 8: Free memory

Step 9: End.



### 30 Merge two arrays

Step-1: Start

Step-2: Input size n1 and n2.

Step-3: Input array 1[n1], array 2[n2].

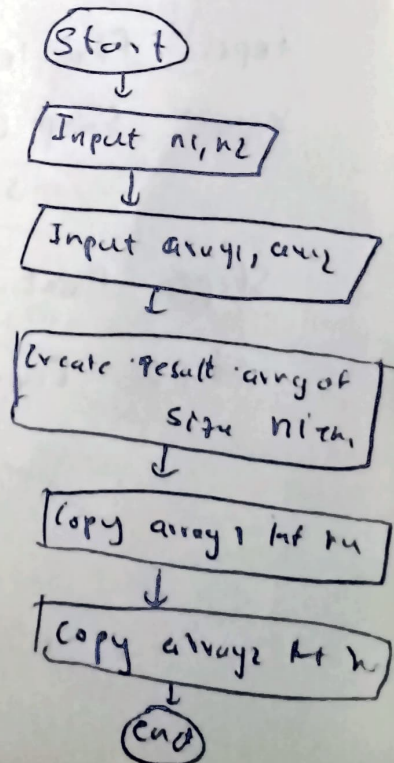
Step-4: Create result array of size n1+n2

Step-5: Copy all elements of array 1 into

Step-6: Copy all elements of array 2

Step-7: Print merged array

Step-8: End



37. check whether a string is palindrome.

Step 1:- start

Step 2:- Input string.

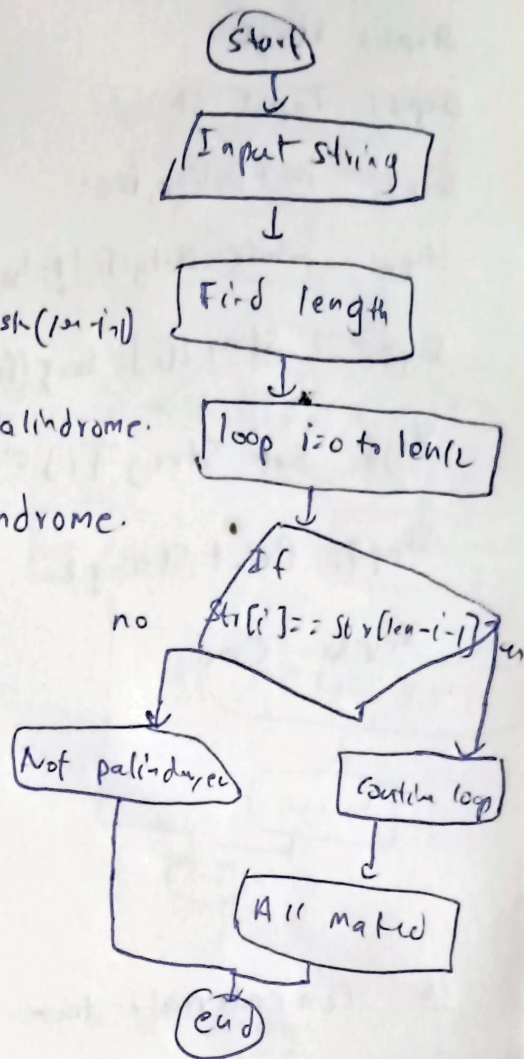
Step 3:- Find length of string.

Step 4:- Compare  $str[i]$  and  $str[len-i-1]$

Step 5:- If mismatch  $\rightarrow$  not palindrome.

Step 6:- If all match  $\rightarrow$  palindrome.

Step 7:- End.



38. Count vowels and consonants in a string.

Step 1:- start

Step 2:- Input string.

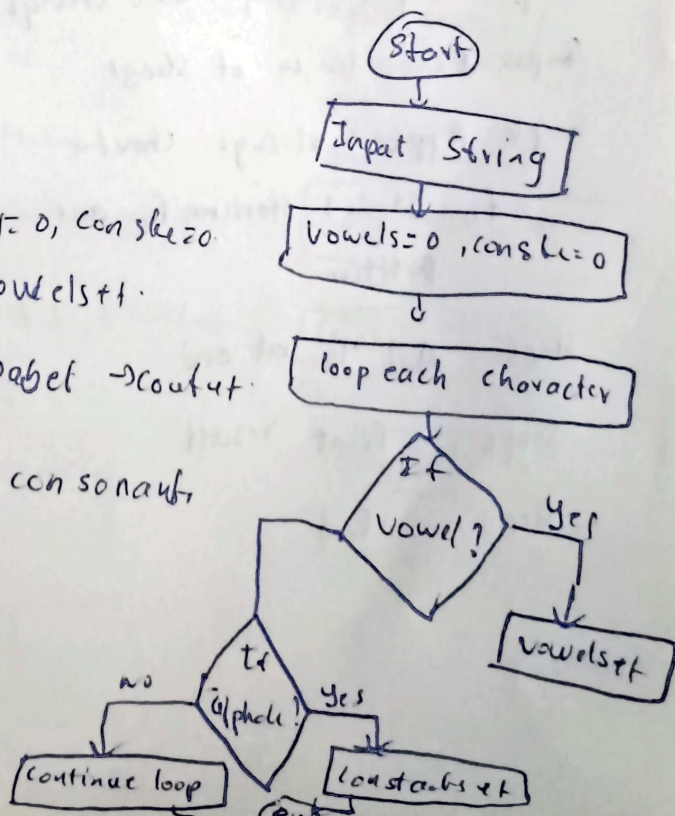
Step 3:- Initialize vowel = 0, cons = 0.

Step 4:- If vowel  $\rightarrow$  vowel++.

Step 5:- Else if alphabet  $\rightarrow$  cons++.

Step 6:- Print vowels, consonants.

Step 7:- End.





3) compare two strings without using strcmp()

Step 1: Start

Step 2: Input string1 and string2

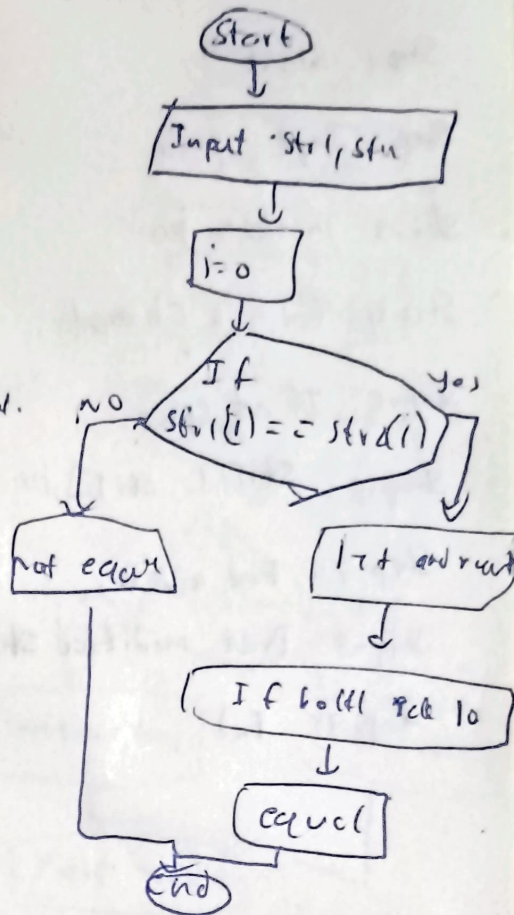
Step 3: Initialize i=0

Step 4: while  $str1[i] == str2[i]$   
and not '\0'.

Step 5: If both end  $\rightarrow$  strings equal.

Step 6: Else  $\rightarrow$  not equal.

Step 7: End.



38. Count number of words in a string

Step 1: Start

Step 2: Input string

Step 3: Initialize count = 0

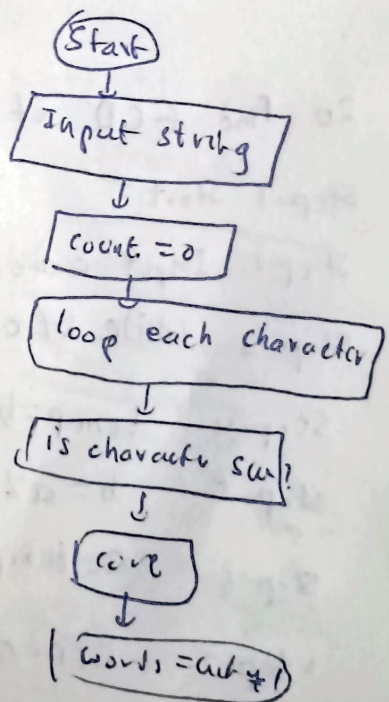
Step 4: Loop through string

Step 5: If space  $\rightarrow$  increment count

Step 6: Total words = count + 1

Step 7: Print result

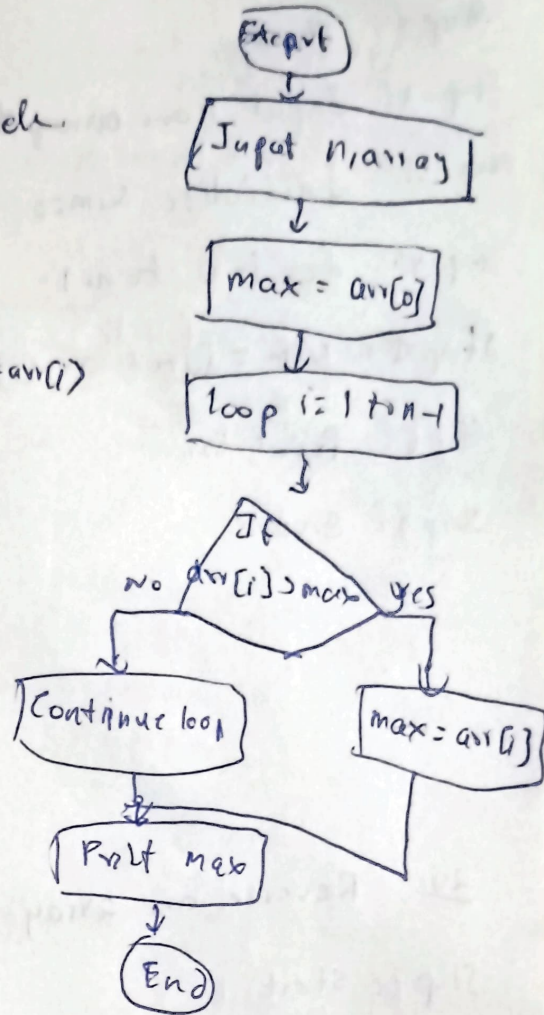
Step 8: End



20 Find longest element in an array.

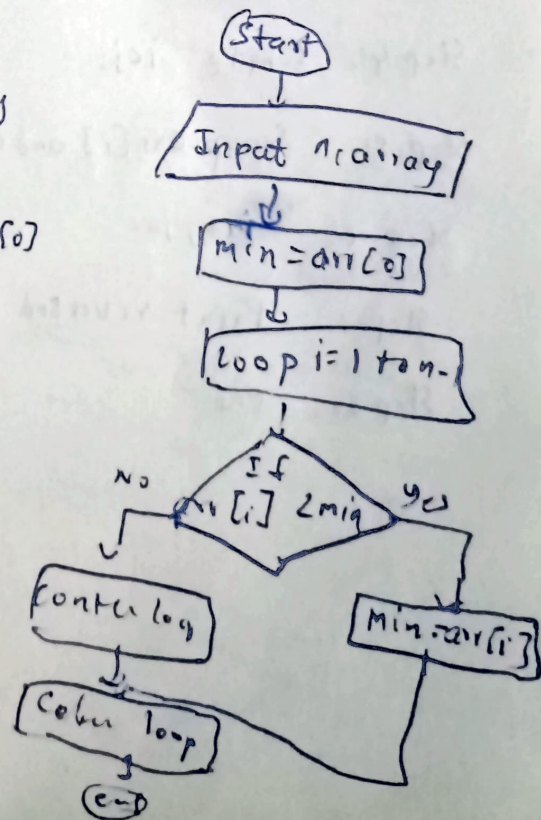
2

- Step-1 Start  
 Step-2 Input size  $n$  and array elements  
 Step-3 Initialize  $\text{max} = \text{arr}[0]$   
 Step-4 for  $i = 1$  to  $n-1$ ;  
 Step-5 If  $\text{arr}[i] > \text{max} \rightarrow \text{max} = \text{arr}[i]$   
 Step-6 Print  $\text{max}$   
 Step-7 end



21 Find smallest element in an array.

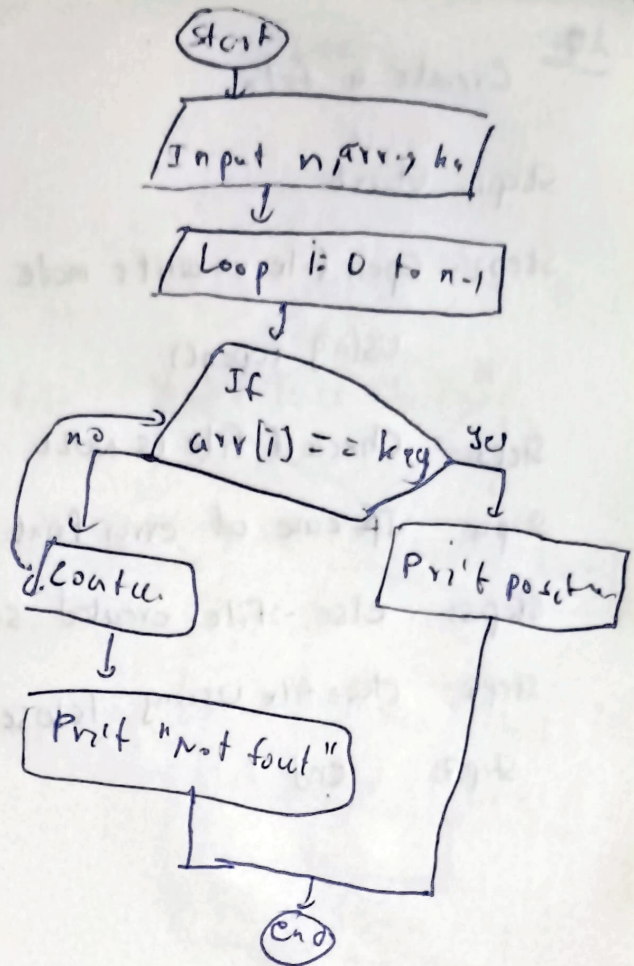
- Step-1 Start  
 Step-2 Input size  $n$  and array elements  
 Step-3 initialize  $\text{min} = \text{arr}[0]$   
 Step-4 for  $i = 1$  to  $n-1$ ;  
 Step-5 If  $\text{arr}[i] < \text{min} \rightarrow$   
 Step-6  $\text{min} = \text{arr}[i]$   
 Step-7 Print  $\text{min}$   
 Step-8 End





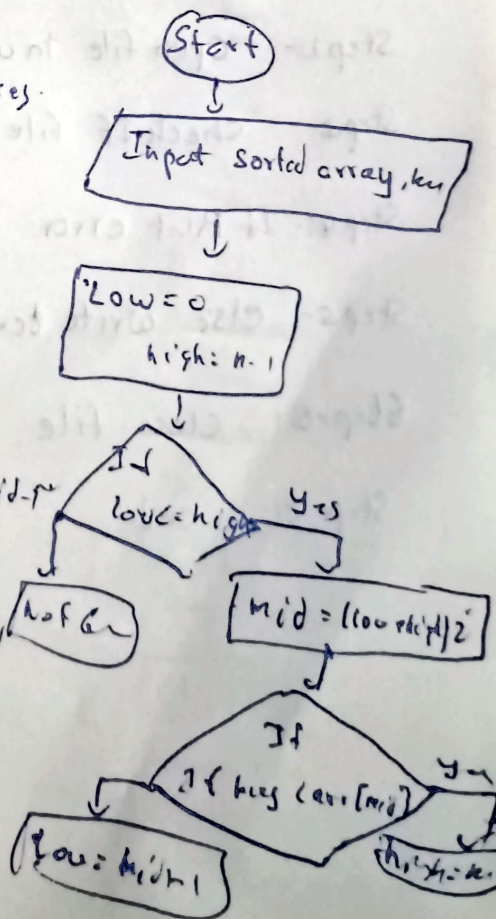
## 1. Linear search

- Step 1: Start  
Step 2: Input  $n$ , array  $arr$ , and key  
Step 3: for  $i = 0$  to  $n-1$ :  
Step 4: If  $arr[i] == key$   
Step 5: Print position  $i$   
Step 6: If not found  
Step 7: Print "Not found"  
Step 8: End



## 2. Binary Search

- Step 1: Start  
Step 2: Input Sorted array  $n$ , and key  
Step 3: Set  $low = 0$ ,  $high = n-1$   
Step 4: while  $low \leq high$ :  
Step 5:  $mid = (low + high) / 2$   
Step 6: If  $arr[mid] == key$  - found  
Step 7: If  $key < arr[mid] \rightarrow high = mid - 1$   
Step 8: else  $low = mid + 1$   
Step 9: If not found - print "Not found"  
Step 10: End



39. Read text from file.

Step1: Start

Step2: Open file is NULL

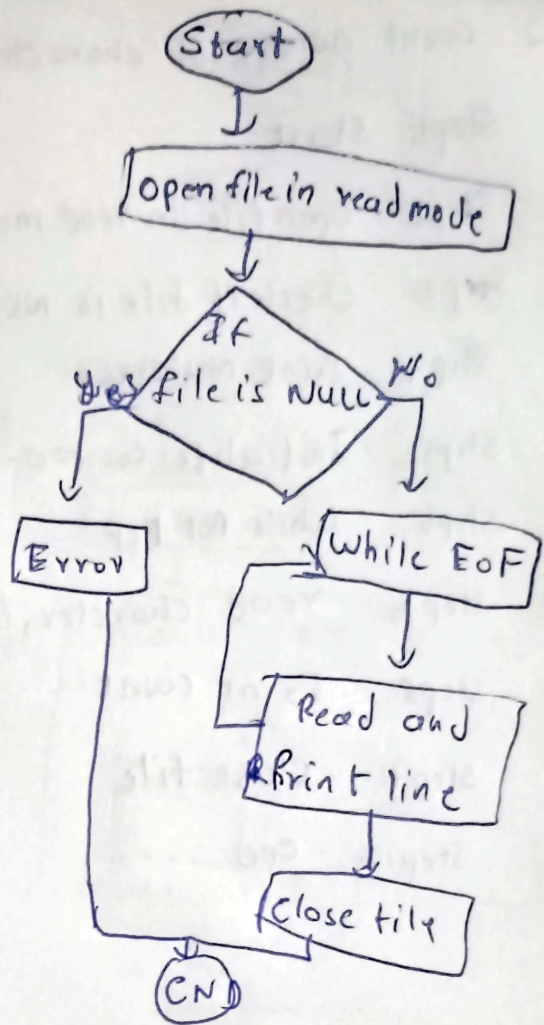
Step3: If Print error

Step4: else while end of file

Step5: read line

Step6: Print each line

Step7: end.



4. Append text to a file.

Step1: Start

Step2: Open file in append mode.

Step3: check if file is NULL.

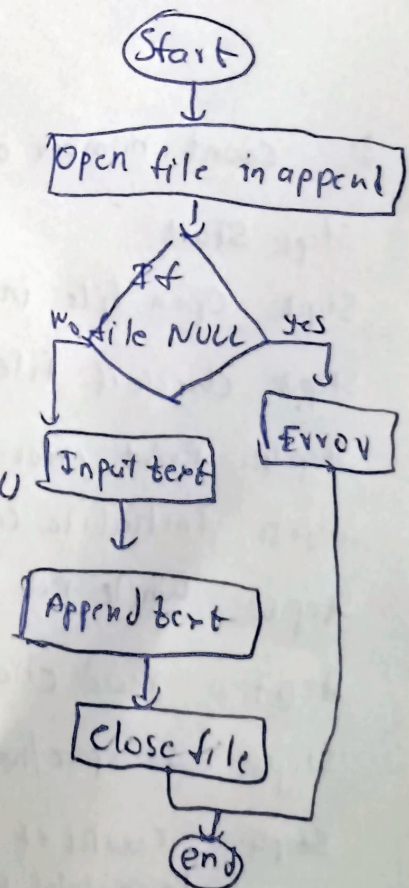
Step4: error.

Step5: Input text from user.

Step6: Write text to file fputs

Step7: Close file

Step8: End





4. Count number of lines in a file

Step 1:- Start

Step 2:- Open file in read mode

Step 3:- Check if file is NULL

Step 4:- Print error

Step 5:- Initialize line count = 0

Step 6:- While not EOF

Step 7:- Read character

Step 8:- If '\n' -> line count++

Step 9:- Print line count

Step 10:- Close file

Step 11:- ENB

