Q4 Sum of array elements using pointers

Step-1 Start

Step 2 Input size of array n

Step-3 Input n elents into arrayarr

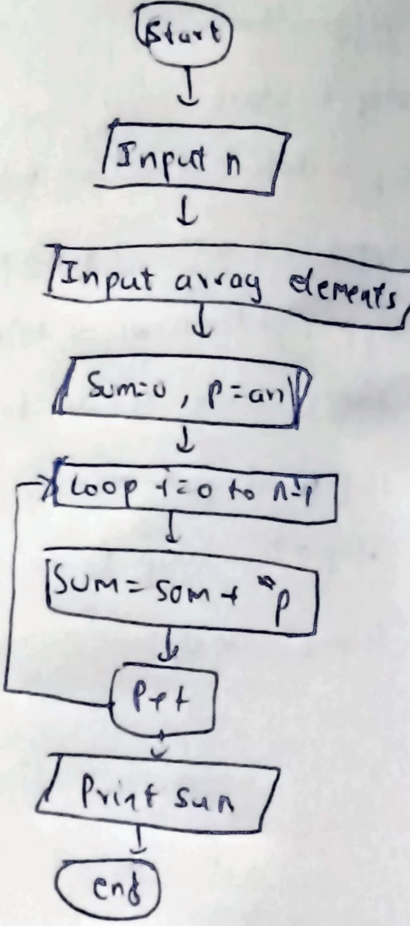
Step-4 Initilis sum=0

Step-5 Use pointer p to travers array

Step-6 Add *p to sum for each element

Step-7 Print sum

Step-8 end



Q5. Reverse a string using recursion.

Step-1 Start

Step-2 Define function reverse(str, i, n)

Step3 If i>= n/2 → return

Step-4 Swap str[i] and str[n-i-1]

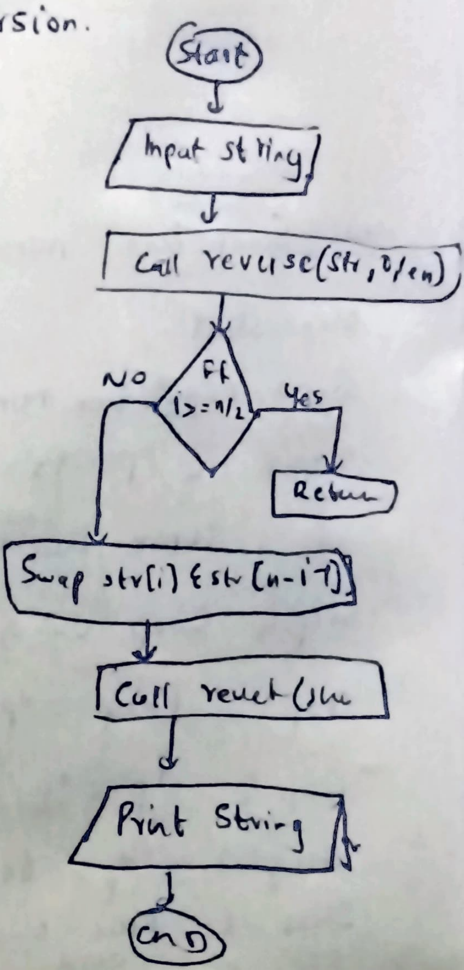
Step5 Call reverse(str, i+1, n)

Step6 Input String.

Step-7 Call reverse (str, 0, length)

Step-8 Print reversed String

Step-9 end



46 Function to check prime

Step1: Start

Step2: Define function is prime(n)

Step3: If n<=1 -> return false

Step4: for i: 2 to n/2 ->

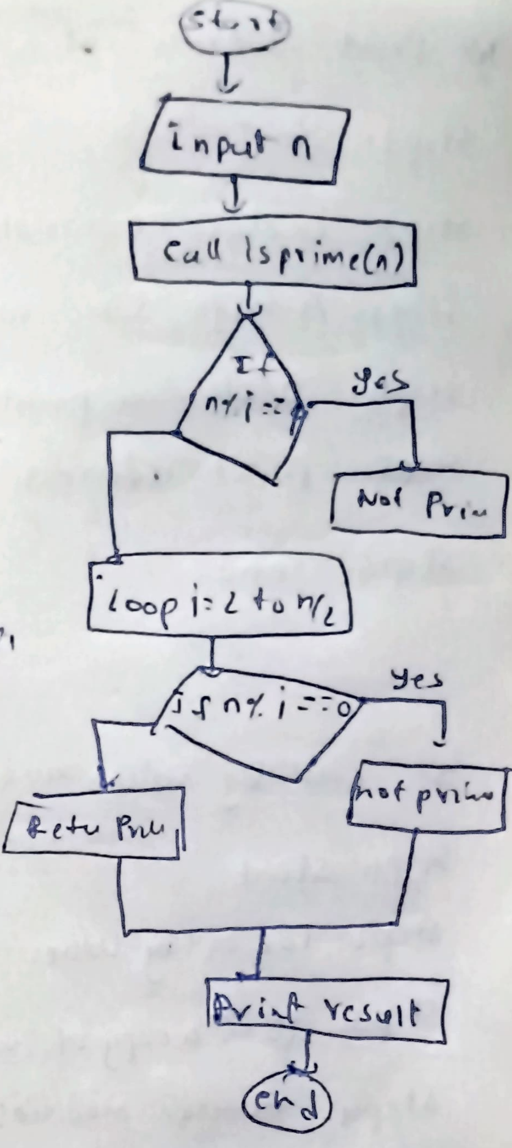
Step5: if n% i== 0 -> return false

Step6: else return true.

Step7: input n.

Step8: Call is Prime(n) and print result

Step9: end



47. Function to return maximum of three numbers

Step-1: Start

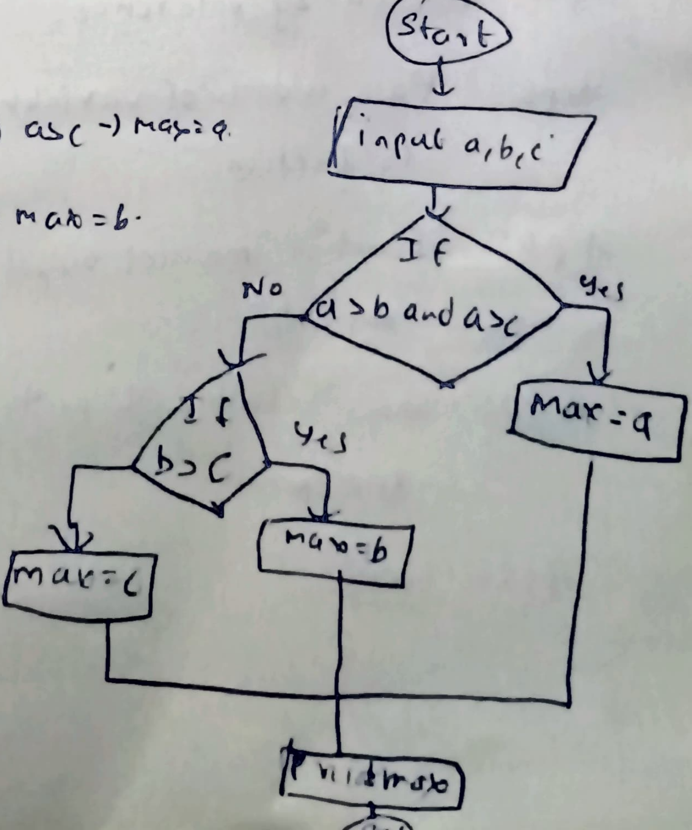
Step-2: Input a,b,c.

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.



48 Print address of a variable using pointer

Step1: Start

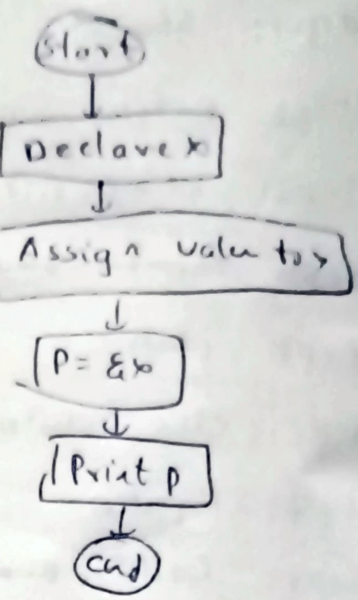
Step2: Declare a variable x.

Step3: Assign some value to x.

Step4: Declare a pointer p= &x

Step5: Print address using p.

Step6: end.



49. call by value and call by reference.

Step1: Start

Step2: For call by value:

Step3: Pass a copy of variabl of fut.

Step4: Function modifies copy, original unchanged.

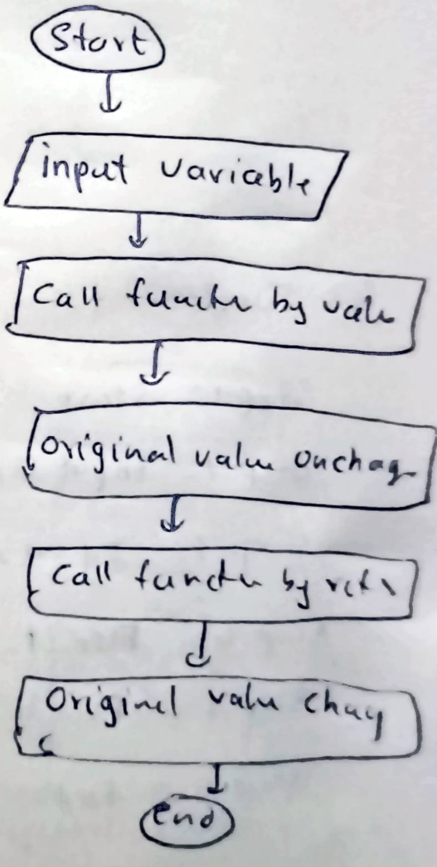
Step5: for call by reference:

Step6. Pass address of variably to fuction.

Step 6: Function modifies origiel variable.

Step 7: Show result with prit statements

Step 8. end.



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

Step1: start

Step2: Input size n

Step3: Allocate memory dynamically
arr = malloc(n*sizeof(int))

Step4: Input n elements into arr

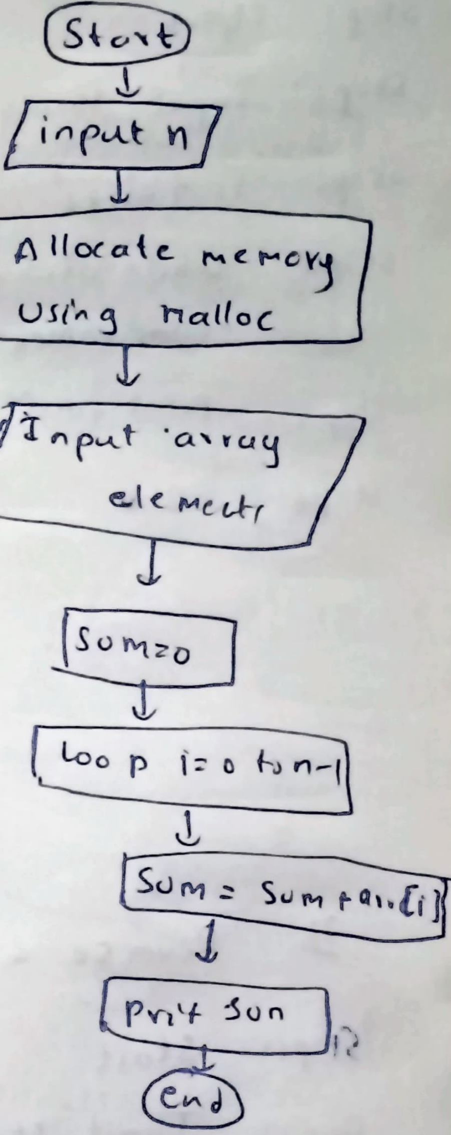
Step5: initialize sum=0

Step6: loop through array + add elements to sum.

Step7: Print sum

Step8: Free memory

Step9: End.



30 Merge two arrays

Step-1: Start

Step-2: Input size n1 and n2.

Step-3: Input array1[n1], array2[n2].

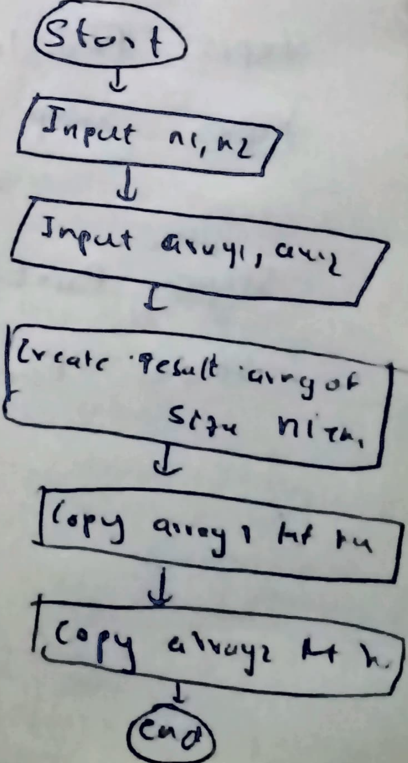
Step-4: Create result array of size n1+n2.

Step-5: Copy all elements of array1 into

Step-6: Copy all elements of arr2

Step-7: Print merged array.

Step-8: End



31 Find length of a string without using strlen().

Step1: Start

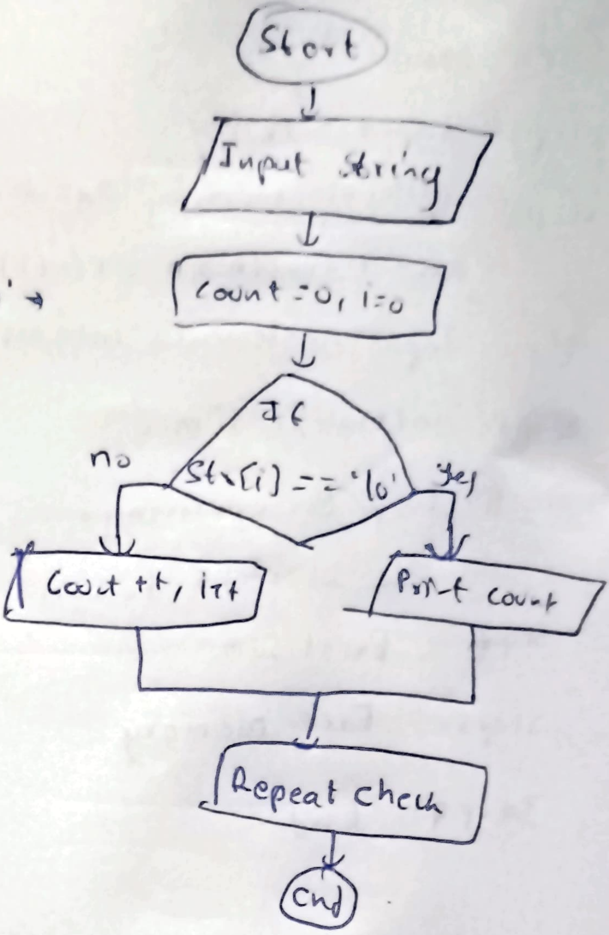
Step2: Input String

Step3: Initialize count=0

Step4: While String[i] != '\0' →
 → increment count

Step5: Print count

Step6: end



32. Reverse a string

Step1: Start

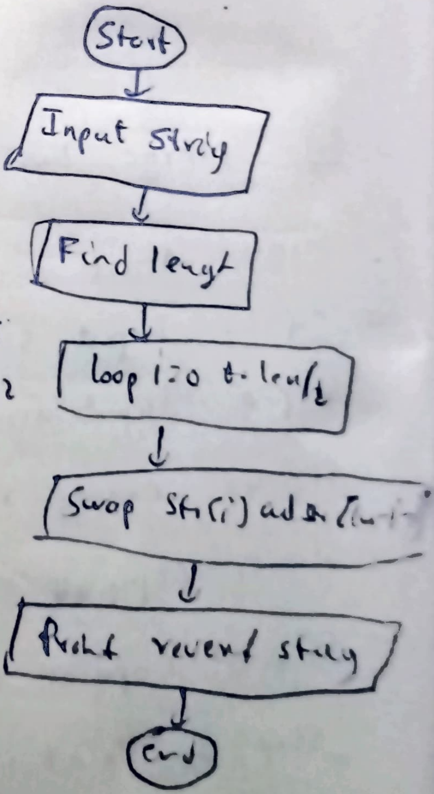
Step2: Input string

Step3: Find length of string.

Step4: Swap Characterise: Str[i] with
Str[len-i-1] Until i < len/2

Step5: Print reversed String.

Step6: End.



2) Check whether a String is palindrome

Step1:- Start

Step2:- Input string.

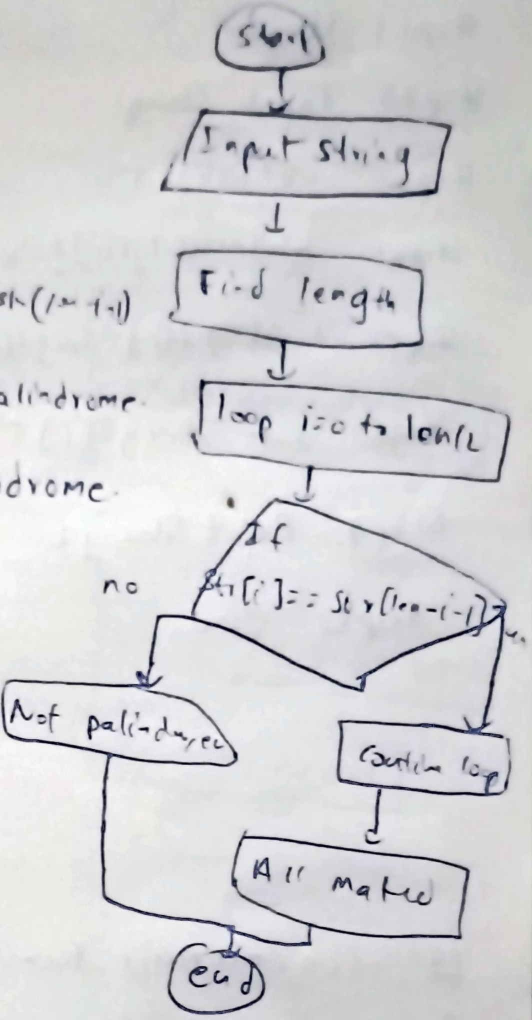
Step3:- Find length of string

Step4:- Compare str[i] and str(len-i-1)

Step5:- If mismatch -> not palindrome.

Step6:- If all match -> Palindrome.

Step7:- End.



3:- Count vowels and consonants in a string.

Step1:- Start

Step2:- Input string.

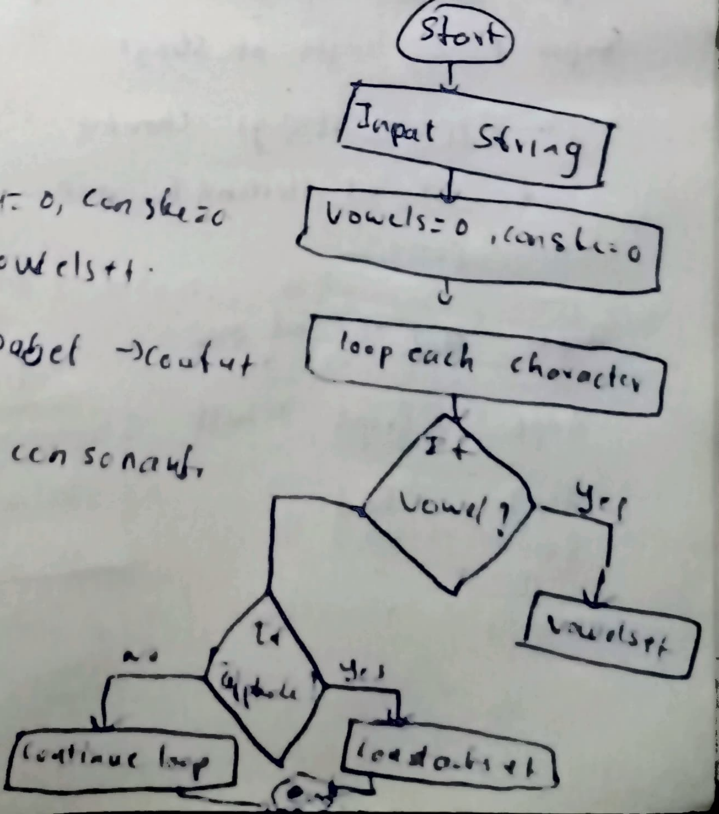
Step3:- Initialize vowel=0, consonants=0

Step4:- If vowel -> vowels++.

Step5:- Else if alphabet -> consonant

Step6:- Print vowels, consonants.

Step7:- End.


Start
Input String
vowels=0, consonants=0
loop each character
If vowel?
yes
vowels++
If alphabet
no
yes
continue loop
consonants++