35. Copy one string to another without using strcpy.

Step 1: Start

Step 2: Input string

Step 3: initialize i=0

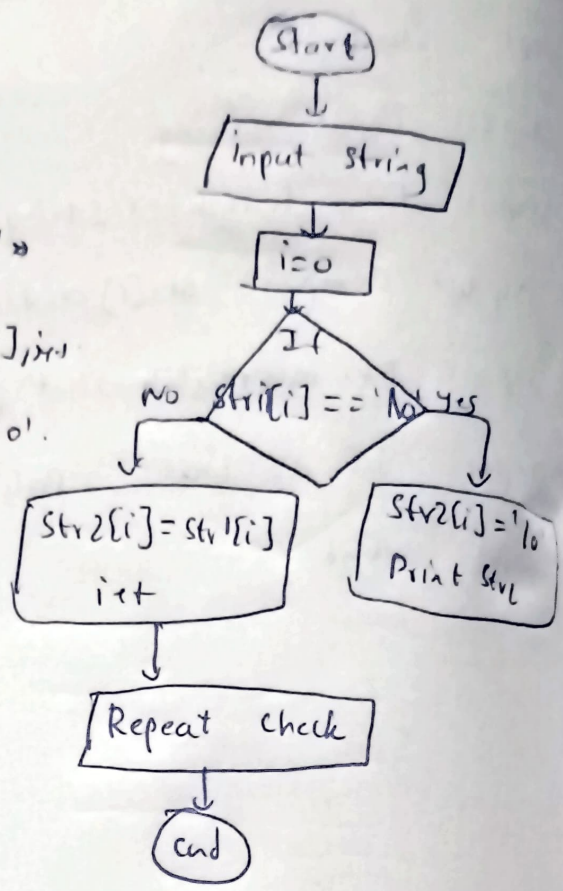
Step 4: While str1[i] != '\0'

Step 5: -> str2[i] = str1[i], i++

Step 6: set str2[i] = '\0'.

Step 7: Print Str2.

Step 8: End



36. Concatenate two strings without using strcat()

Step 1: Start

Step 2: Input string1 and string2.

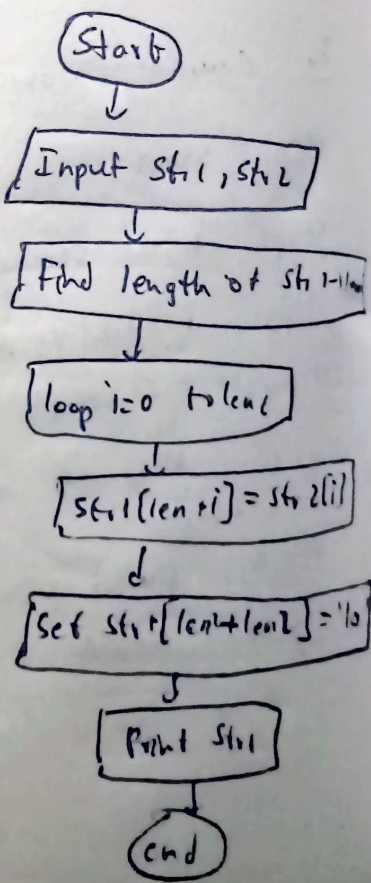
Step 3: Find length of string1

Step 4: Append string2 char to string1 starting from at position

Step 5: Add '\0' at end

Step 6: Print result

Step 7: End.



3) Compare two strings without using strcmp()

Step1: Start

Step2: Input string1 and string2

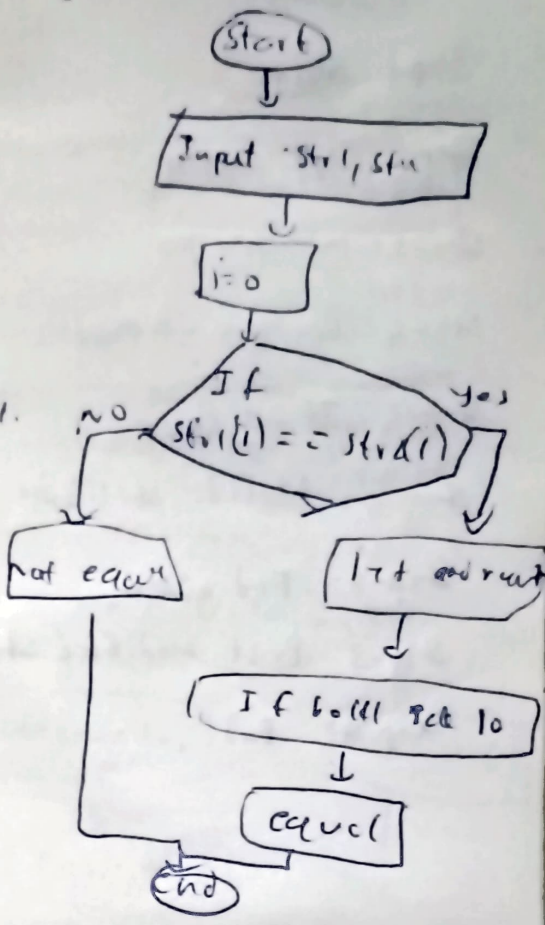
Step3: Initialize i=0

Step4: while str1[i] == str2[i]
and not '\0'

Step5: If both end → strings equal.

Step6: Else → not equal.

Step7: End.



35. Count number of words in a string

Step1: Start

Step2: Input string

Step3: Initialize count = 0.

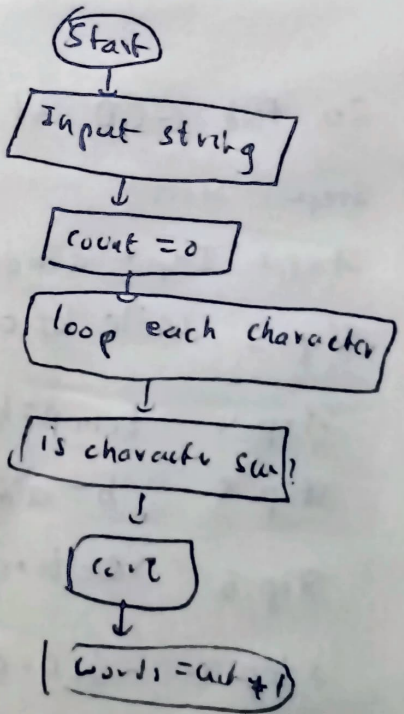
Step4: Loop through string

Step5: If space → increment count.

Step6: Total words = count + 1.

Step7: Print result

Step8: End



19 Remove all spaces from a string

Step-1 Start

Step-2 Input string

Step-3 Initialize j=0.

Step-4 For each character:

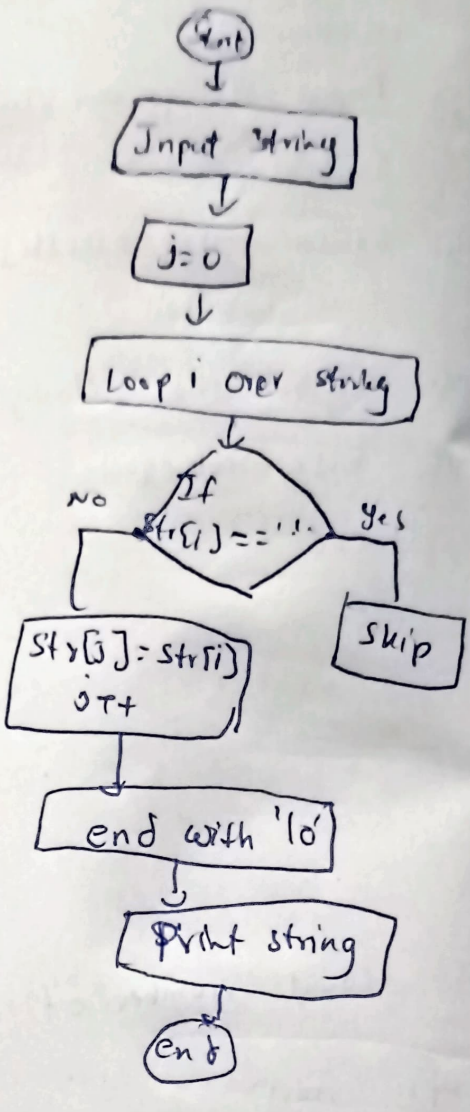
Step-5 If not space

Step-6 str[j] = str[i], j++

Step-7 End with '\0'.

Step-8 Print modified string.

Step-9 End



20 Find GCD of two numbers.

Step-1 Start

Step-2 Input a and b

Step-3 while b ≠ 0

Step-4 temp = b

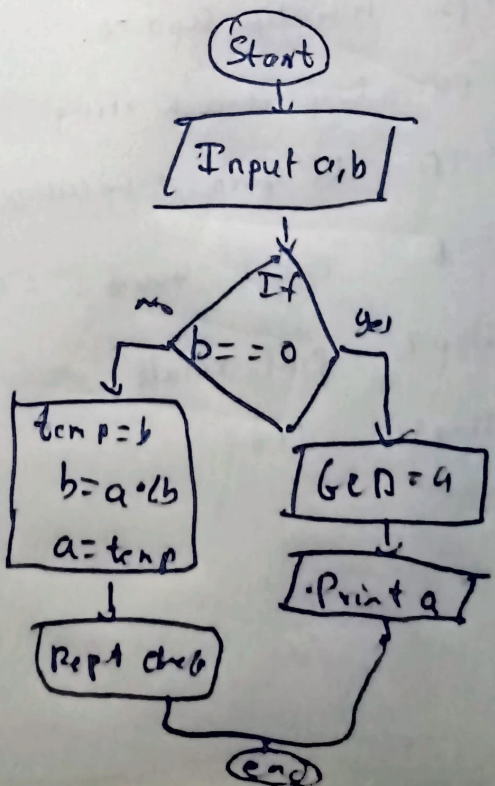
Step-5 b = a % b

Step-6 a = temp

Step-7 GCD = a

Step-8 Print GCD

Step-9 End



19 Remove all spaces from a string

Step 1 Start

Step 2 Input string

Step-3 initialize j=0

Step 4: for each character:

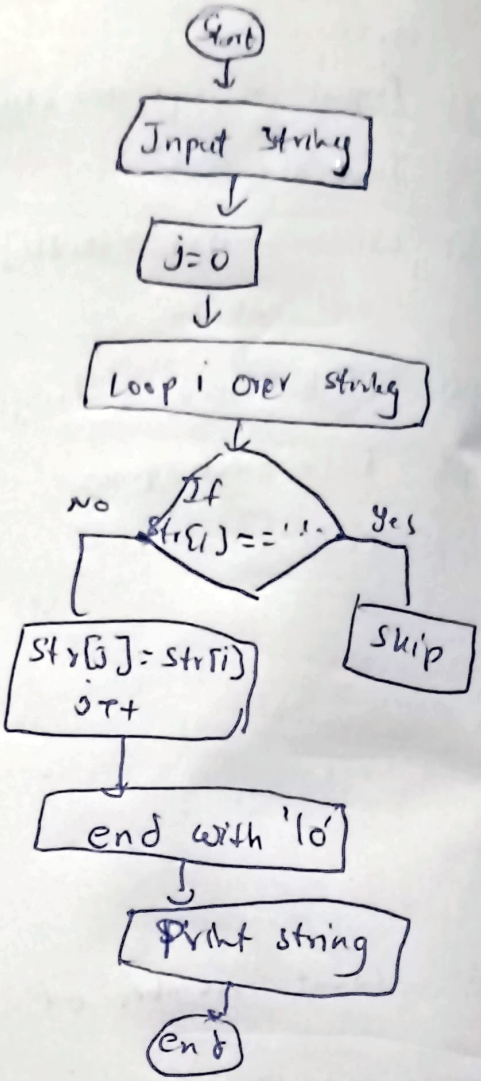
Step 5 If not space

Step-6: str[j] = str[i], j++

Step-7: End with '\0'.

Step-8: Print modified string

Step-9: End



20 Find GCD of two numbers.

Step-1 Start

Step-2 Input a and b

Step-3 while b != 0

Step-4 temp = b

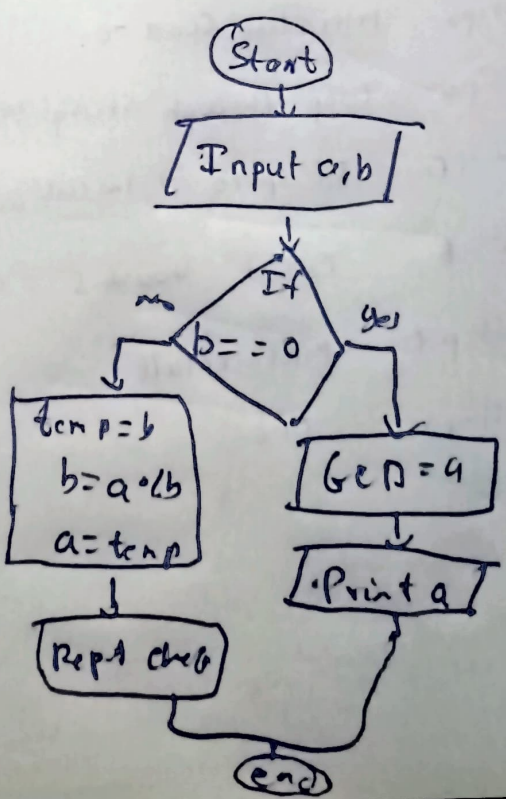
Step-5 b = a % b

Step-6 a = temp

Step-7 GCD = a

Step-8 Print GCD

Step 9 End



2. Find largest element in an array.

Step-1 Start

Step-2 Input size n and array elements

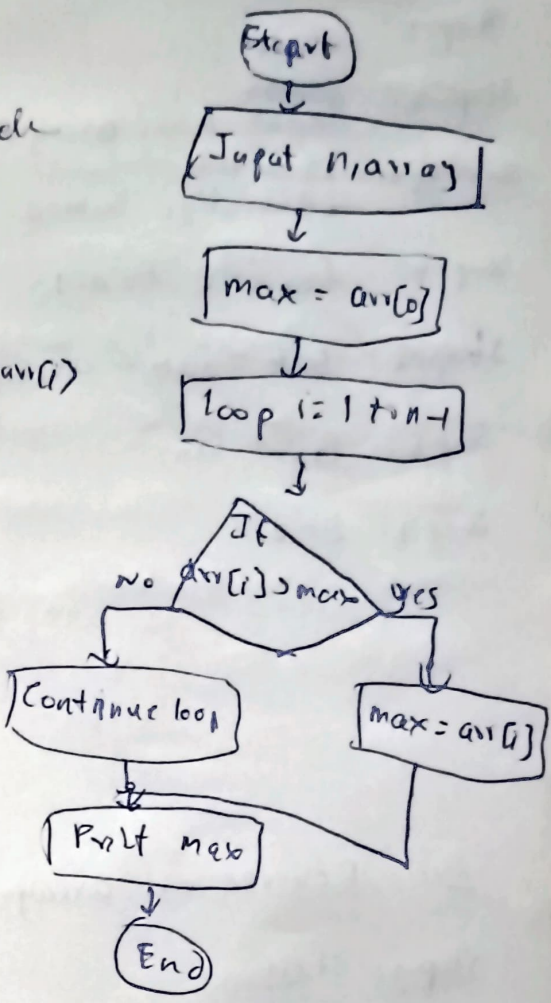
Step-3 Initialize max = arr[0]

Step-4 for i = 1 to n-1;

Step-5 If arr[i] > max → max = arr[i]

Step-6 Print 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 min = arr[0]

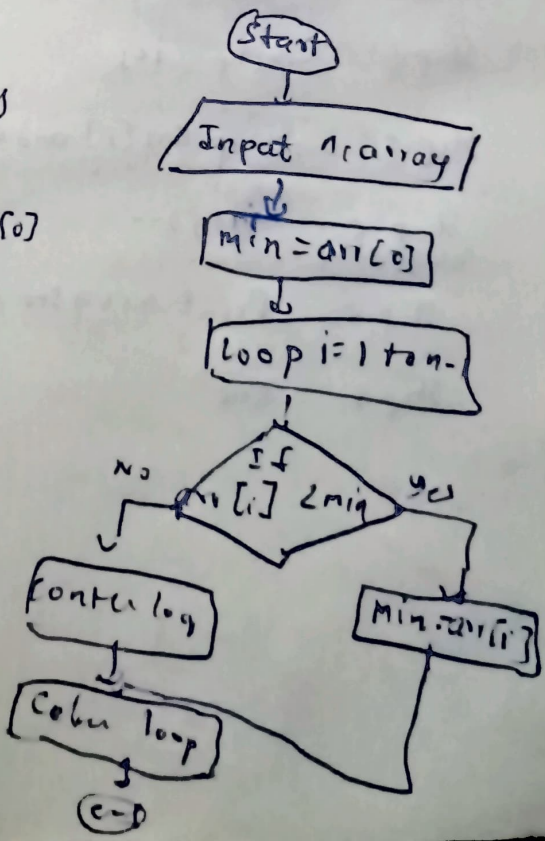
Step-4 for i = 1 to n-1:

Step-5 If arr[i] < min →

Step-6 min = arr[i]

Step-7 Print min

Step-8 End



2: Find sum of array elements.

Step-1: Start

Step-2: Input n and array elements

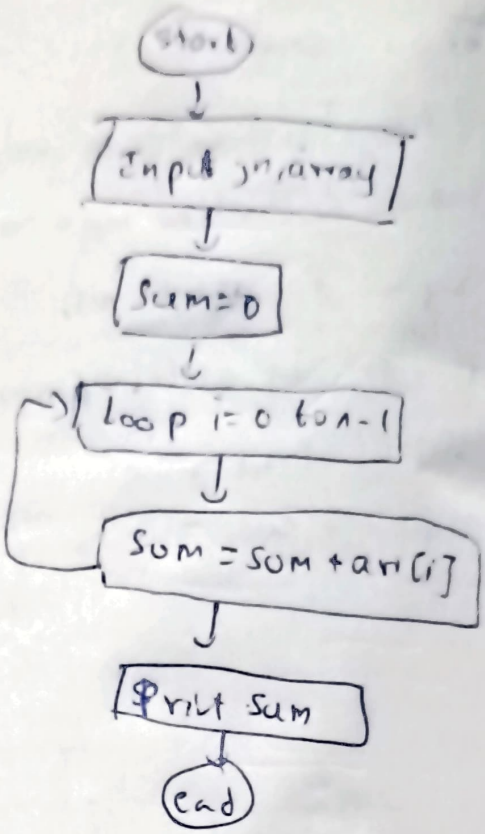
Step-3: Initialize sum=0

Step-4: for i=0 to n-1

Step-5: sum = sum + arr[i]

Step-6: Print sum

Step-7: End



20: Reverse an array

Step-1: Start

Step-2: Input n and array elements

Step-3: Initialize i=0, j=n-1

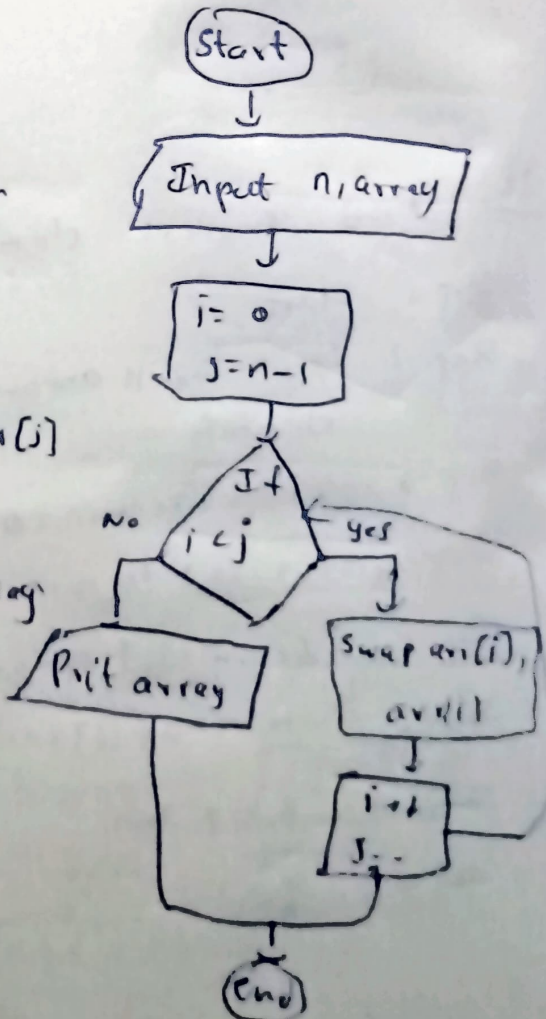
Step-4: while i<j:

Step-5: Swap arr[i] and arr[j]

Step-6: i++, j--

Step-7: Print reversed array

Step-8: End



1. Linear search

Step 1: start

Step 2: Input n, array (arr), key

Step 3: for i: 0 to n-1:

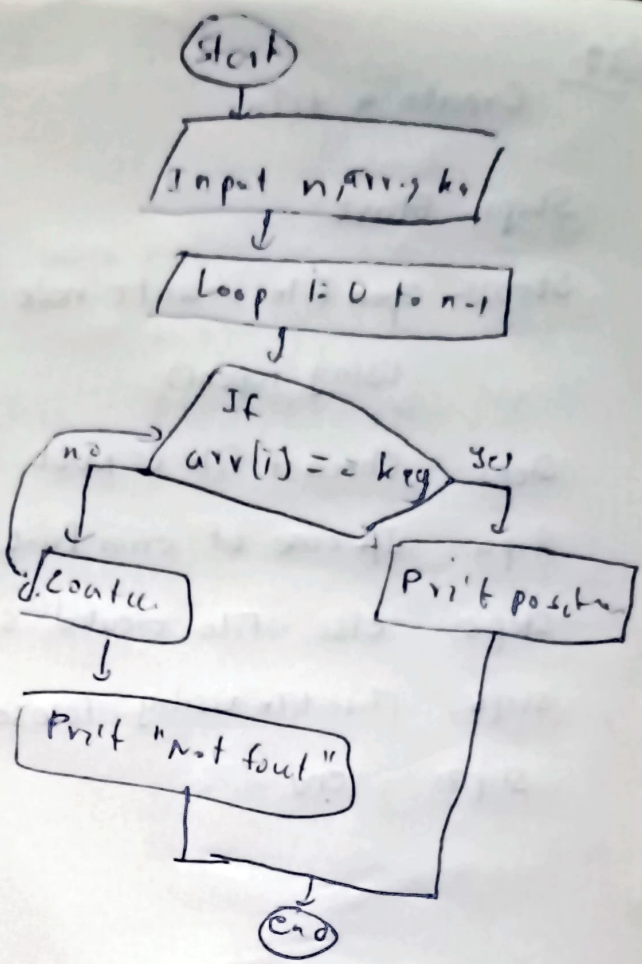
Step 4: If arr[i] = = key

Step 5: Print position of

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 ≤ high:

Step 5: mid = (low + high) / 2

Step 6: If arr[mid] = = key → found

Step 7: If key < arr[mid] → high = mid - 1

Step 8: else low = mid + 1

Step 9: If not found → print "Not found"

Step 10: End