

Q1 Reverse the array. i/b → {1,2,33 0/b → {3,2,13

The approach of the above question can be of 2 pointer approach. Take i = 0 & j = size-1, Swap them & increment the value of i and decrement the value of i and decrement the value of j. This will go on until i < = 1.

Dry run on away of characters

1) S = {'h', 'e', 'l', 'l', 'o'}

l'=0, j'=4

In this case as sosize () -1 = 4 here

i < = j - True

Swap (SCiJ, SCjJ)

1°++

j - - i

2) S = { 'o', 'e', 'l', 'l', 'h'}

l°=1, j=3

e'<=jor)True

Swap (S[i], S[j])

1++

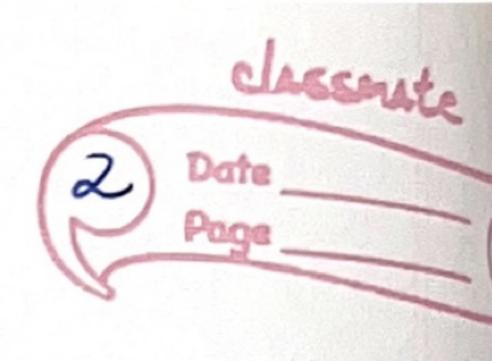
1 --

3) S = { 'O', 'l', 'l', 'e', 'e', 'h''}

i=2,j=2

i < = j - True

Swap (S[i], S[j])



1++ { (0), (l), (l), (e), (h) 4

Here it is hence the condition is false.

Hence here we have got the reversed array.

void reverse String (vector < char) 4 s) {

int i = 0 50 //2First lindexed, in 5 = 2 int i = S. size () - 1 i // Last index // Running loop

while (i < = -i) {

swap (Stij) sij/Swap i++ i // Increment the value of it j-- i// Decrement the value of j

The above code can be run for integer arrays also. Just use vector of integer.

* Time Complexity = O(n)

* Space Complexity = O(1) → no extra space is

F. 43, (3), (1), (2), (0)