Given an array of only 0's & 1's Sort the array in ascending order.

l = 0 , r = 5 - 1

i/p → arr = {1, 1, 0, 0, 0, 0, 1, 0};

(left at the first element, right at the last)

Two pointer approach:

While arr[left] == 1
arr[right] == 0
swap (left, right)
l++, r--

While (arr[left] == 0, & left < right)
left++

While (arr(right) == 1, & left < right)
right --

left	l	l		r	r	r	right
1	1	0	0	0	0	1	0
0	0				1		1

l = 0, 1
r = 7, 6
l < r

Kadane's Algorithm : → O(n)

Maximum Subarray Sum

arr = { 5, -8, 1, 2, -1, 4 }
(indices 0, 1, 2, 3, 4, 5)

cmax = arr[0] → 5, -3, 1, 3
gmax = arr[0] → 5, 6

```
for (int i = 1; i < n; i++) {
    cmax = max (arr[i], cmax + arr[i]);
    gmax = max (cmax, gmax);
}
return gmax;
```

arr[i], cm + arr[i]
6: -8, 5 + (-8) = -3
2: 1, -3 + 1 = -2
2, 1 + 2 = 3
-1, 3 + (-1) = 2
4, 6

Character Arrays & Strings : → scanf( ) → gets( ), puts( )

Definition : → A string is a sequence / collection of characters. It can contain :

1. Alphabets → 'A' to 'Z' or 'a' to 'z'
2. Numbers → '0' – '9'
3. special symbols → #, /, \, *, etc etc
4. Whitespaces

strcpy
strcmp
strlen
strconcat

"JIT" → 'J', 'I', 'T', '\0' (end of string)

#include <string.h>

* Program to print the first letter of each word :

Rahul Is From AIML Batch 1 '\0'

(indices 0–25; arrows mark the spaces; '\0' — not a letter, null character)

O/p : → R I F A B

* for the index 0 element → if it's an alphabet, we will print it or else we won't.

```
while (str[i] != '\0') {
    if (str[i] == ' ' && str[i+1] → alpha
        print str[i];
}
```

str1 = Java   str2 = Python

swap (str1, str2)

str1 = Python   str2 = Java

MALAYALAM (start → M, end → M)   (RaDaR) → ctype → tolower( )

```
while (start < end) {
    if (str(start) != str(end))
        return 0;
    s++
    e--
}
```

Language Expertise : C, C++, Java, Python
SQL, Mongo dB,
NoSQL

C, C++
↳ Java
(python) → DTL

print(type(a))   a = 10;
a = 20.5
<class float>

* Remove duplicates from a String

index = 0, 1, 2, 3

SSAURAV'S → SAURV'

```
for (i = 0; i < len; i++) {
    int j;
    for (j = 0; j < i; j++) {
        if (str[i] == str[j]) {
            break;
        }
    }
    if (i == j) → str[index++] = str[i];
```

j == i

str = SAURV'

DRY RUN