

## L10A : Test 2

### Q1 : Print 2D Array

[Send Feedback](#)

Given a 2D integer array with n rows and m columns. Print the 0th row from input n times, 1st row n-1 times.....(n-1)th row will be printed 1 time.

#### Input format :

Line 1 : No of rows (n) and no of columns (m) (separated by single space)

Line 2 : Row 1 elements (separated by space)

Line 3 : Row 2 elements (separated by space)

Line 4 : and so on

#### Sample Input 1:

```
3 3
1 2 3
4 5 6
7 8 9
```

#### Sample Output 1 :

```
1 2 3
1 2 3
1 2 3
4 5 6
4 5 6
7 8 9
```

```
1. #include <iostream>
2. using namespace std;
3.
4. void print2DArray(int **input, int row, int col) {
5.     // Write your code here
6.     for(int i = 0; i < row; i++){
7.
8.         for(int k = 0; k < row-i; k++){
9.             for(int j = 0; j < col ; j++){
10.                 cout << input[i][j] << " ";
11.             }
12.             cout << endl;
13.         }
14.
15.     }
16. }
```

## Q2 : Minimum Length Word

[Send Feedback](#)

Given a string S (that can contain multiple words), you need to find the word which has minimum length.

**Note : If multiple words are of same length, then answer will be first minimum length word in the string.**

**Words are separated by single space only.**

**Input Format :**

String S

**Output Format :**

Minimum length word

**Constraints :**

$1 \leq \text{Length of String S} \leq 10^5$

**Sample Input 1 :**

this is test string

**Sample Output 1 :**

is

**Sample Input 2 :**

abc de ghijkl a uvw h j

**Sample Output 2 :**

a

```
1.  /* input - Input String
2.  * output - Save the result in the output array (passed as argument). You don't have to
3.  * print or return the result
4.  */
5.  #include<bits/stdc++.h>
6.  #include<string>
7.  #include<cstring>
8.  void minLengthWord(char input[], char output[]){
9.
10.     // Write your code here
11.     int min = 1e6;
12.     int si = 0,ei = 0,i = 0;
13.     int k = 0;
14.     int n = 0;
15.     while(input[i] != '\0'){
16.         n++;
17.         i++;
18.     }
19.     i = 0;
20.
21.
22.     while(i < n){
23.
```

```
24.     if(input[i] == ' '){
25.         ei = i;
26.         int curlen = ei - si;
27.         if(curlen < min){
28.             min = curlen;
29.             k = 0;
30.             while(si < ei){
31.                 output[k++] = input[si];
32.                 si++;
33.             }
34.             if(input[si] == ' '){
35.                 si++;
36.             }
37.
38.         }else if(curlen >= min){
39.             si = ei + 1;
40.         }
41.
42.
43.     }
44.
45.
46.     i++;
47. }
48.
49.     ei = n;
50.     int minlast = ei-si;
51.     if(minlast < min){
52.         min = minlast;
53.         k = 0;
54.         while(si < ei){
55.             output[k++] = input[si];
56.             si++;
57.         }
58.     }
59.
60.
61.     output[k] = '\0';
62.
63.
64. }
```

### Q3 : Leaders in array

[Send Feedback](#)

Given an integer array A of size n. Find and print all the leaders present in the input array. An array element A[i] is called Leader, if all the elements following it (i.e. present at its right) are less than or equal to A[i].

Print all the leader elements separated by space and in the same order they are present in the input array.

#### Input Format :

Line 1 : Integer n, size of array

Line 2 : Array A elements (separated by space)

#### Output Format :

leaders of array (separated by space)

#### Constraints :

$1 \leq n \leq 10^6$

#### Sample Input 1 :

```
6
3 12 34 2 0 -1
```

#### Sample Output 1 :

```
34 2 0 -1
```

#### Sample Input 2 :

```
5
13 17 5 4 6
```

#### Sample Output 2 :

```
17 6
```

```
1. void Leaders(int* arr,int len)
2. {
3.     /* Don't write main().
4.     * Don't read input, it is passed as function argument.
5.     * Print your output exactly in the same format as shown.
6.     * Don't print any extra line.
7.     */
8.     for(int i = 0; i < len; i++){
9.         int flag = 1;
10.        int curr_leader = arr[i];
11.
12.        for(int j = i; j < len; j++){
13.            if(arr[j] > curr_leader){
14.                flag = 0;
15.                break;
16.            }
17.        }
18.        if(flag == 1){
```

```
19.         cout << curr_leader << " ";
20.     }
21.
22.
23.     }
24. }
25.
26.
```