**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

#include <iostream>

#include "solution.h"

using namespace std;

void print2DArray(int \*\*input, int row, int col) {

// Write your code here

int k = row;

for (int i = 0; i < row; i++)

{

for (int l = k - 1; l >= 0; l--)

{

for (int j = 0; j < col; j++)

{

cout << input[i][j] << " ";

}

cout << "\n";

}

k--;

}

}

int main() {

int row, col;

cin >> row >> col;

int \*\*input = new int\*[row];

for(int i = 0; i < row; i++) {

input[i] = new int[col];

for(int j = 0; j < col; j++) {

cin >> input[i][j];

}

}

print2DArray(input, row, col);

}

**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 <= n <= 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

#include<iostream>

#include<climits>

using namespace std;

#include"solution.h"

int main()

{

int len;

cin>>len;

int \*arr = new int[len + 1];

for(int i=0;i<len;i++)

{

cin>>arr[i];

}

Leaders(arr,len);

}

**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 seperated by single space only.**

**Input Format :**

String S

**Output Format :**

Minimum length word

**Constraints :**

1 <= Length of String S <= 10^5

**Sample Input 1 :**

this is test string

**Sample Output 1 :**

is

**Sample Input 2 :**

abc de ghihjk a uvw h j

**Sample Output 2 :**

a

#include <limits.h>

#include <string.h>

void minLengthWord(char input[], char output[])

{

int minStart = -1;

int minLength = INT\_MAX;

int currentStart = 0;

int i = 0;

int len = strlen(input);

for (; i < len; i++)

{

if (input[i] == ' ')

{

int currentWordLength = i - currentStart;

if (currentWordLength < minLength)

{

minStart = currentStart;

minLength = currentWordLength;

}

currentStart = i + 1;

}

}

if (minStart == -1)

{

for (int i = 0; i <= len; i++)

{

output[i] = input[i];

}

}

else

{

int currentWordLength = i - currentStart;

if (currentWordLength < minLength)

{

minStart = currentStart;

minLength = currentWordLength;

}

int j = 0;

for (int i = minStart; i < minStart + minLength; i++)

{

output[j++] = input[i];

}

output[j] = '\0';

}

}