

Computer Professionals Program
Master of Science in Computer Science Cooperative Program

Applicant Information Form

Subhra Dey logged on
Applicant ID: 381763

This exam and all related materials are private copyrighted material and the intellectual property of M.I.U. and are provided solely for individual use on this admissions pre-test. They are not to be copied or further distributed without written permission from M.I.U.

The purpose of this short test is to assess your ability to solve elementary programming problems in a language of your choice. Write your solutions in Java if you are familiar with that language; otherwise use one of these languages: C, C++, or C#. If you do not have access to a compiler for your language, write your answers in a text editor such as notepad and mention in a comment that you did not use a compiler.

For each of the problems below, write the simplest, clearest solution you can, in the form of a short program. Answer as much as you can for a problem, even if you do not have the complete answer.

If you are using C or C++ and the function you are writing requires an array parameter then you will also have to have a parameter that is the length of the array. This is not necessary in C# or Java since an array is an object in those languages and has a length method that returns the length of the array.

You do not need to do any I/O, i.e., you can hard-code your input data and do not have to write out anything to the console. Keep it simple! We are primarily interested in what you write in the body of the function. However, please be sure that your solution will work for all valid input data.

The clock is ticking now, so you don't have time to ask for clarifications on any of the questions. If something is not clear to you, resolve it yourself and state in a comment in the program what was unclear and how you resolved it.

When you have finished an answer, copy and paste it into the text box associated with the question and click the submit button to save it in our database. If you change an answer and submit it again, the previous version of the answer will be overwritten with the new version.

1. An array is defined to be **stepped** if it is in ascending order and there are 3 or more occurrences of each distinct value in the array. Note that ascending order means $a[n] <= a[n+1]$. It does not mean $a[n] < a[n+1]$ strictly ascending). Write a function named *isStepped* that returns 1 if its array argument is stepped; otherwise return 0.

If you are programming in Java or C#, the signature is

`int isStepped(int[] a)`

If you are programming in C or C++, the signature is

`int isStepped(int a[], int len)` where len is the number of elements in the array.

Examples

If the array is	return	reason
{1, 1, 1, 5, 5, 5, 5, 8, 8}	1	It is in ascending order. The distinct values of the array are 1, 5, 8 and there are three or more occurrences of each of these values.
{1, 1, 5, 5, 5, 5, 8, 8}	0	Even though it is in ascending order, there are only two occurrences of the value 1.
{5, 5, 5, 15}	0	Even though it is in ascending order, there is only one occurrence of the value 15.
{3, 3, 3, 2, 2, 2, 5, 5, 5}	0	It is not in ascending order
{3, 3, 3, 2, 2, 2, 1, 1, 1}	0	It is not in ascending order
{1, 1, 1}	1	It is in ascending order and there are three or more occurrences of each distinct value. In this case there is only one distinct value.
{1, 1, 1, 1, 1, 1, 1}	1	It is in ascending order and there are three or more occurrences of each distinct value. In this case there is only one distinct value.

Copy and paste your answer here and click the "Submit answer" button

Submit answer to question 1

You should see a confirmation popup after hitting the submit button above.

2. An **onion array** is an array that satisfies the following condition for all values of j and k:

$j >= 0 \text{ and } k >= 0 \text{ and } j+k <= \text{length of array} \text{ and } j < k \text{ then } a[j]+a[k] <= 10$

Write a function named *isOnionArray* that returns 1 if its array argument is an onion array and returns 0 if it is not.

Your solution must not use a nested loop (i.e., a loop executed from inside another loop). Furthermore, once you determine that the array is not an onion array your function must return 0; no wasted loops cycles please!

If you are programming in Java or C#, the function signature is

`int isOnionArray(int[] a)`

If you are programming in C or C++, the function signature is

`int isOnionArray(int a[], int len)` where len is the number of elements in the array a.

Examples

a is	then function returns	reason
{1, 2, 19, 4, 5}	1	because 1+5 <= 10, 2+4 <=10
{1, 2, 3, 4, 15}	0	because 1+15 > 10
{1, 3, 9, 8}	0	because 3+9 > 10
{2}	1	because there is no j, k, where $a[j]+a[k] > 10$ and $j+k = \text{length of array}$ and $j < k$
{}	1	because there is no j, k, where $a[j]+a[k] > 10$ and $j+k = \text{length of array}$ and $j < k$
{-2, 5, 0, 5, 12}	1	because -2+12 <= 10 and 5+5 <= 10

Copy and paste your answer here and click the "Submit answer" button

Submit answer to question 2

You should see a confirmation popup after hitting the submit button above.

3. An **Olympic array** is defined to be an array in which every value is greater than or equal to the sum of the values less than it. The sum of the values less than the minimum value in the array is defined to be 0.

For example, {3, 2, 1} is an Olympic array because

- a. 1 is the minimum value and by definition the sum of the values less than it is 0. Since 1 is greater than 0, it satisfies the condition.
- b. There is only one value less than 2 and 2 is greater than it, so the value 2 satisfies the condition.
- c. The values 1 and 2 are less than 3 and 3 is equal to their sum, so the value 3 satisfies the condition.

Hence all elements of the array satisfy the conditions and the array is an Olympic array.

{2, 2, 1, 1} is also an Olympic array because the values less than 2 sum to 2.

{1, 1000, 100, 10000, 2} is also an Olympic array. However, {1, 99, 99, 1000, 100, 10000, 2} is not an Olympic array because the sum of the numbers less than 100 (99+99+1) is greater than 100. Please be sure that your function detects that this is not an Olympic array!

{1, 2, 1, 3, 2} is not an Olympic array because 3 is not greater than or equal to $1+2+1+2$.

{1, 2, -1, 2, 2} is not an Olympic array because -1 is the minimum value but it is not greater than or equal to 0.

Write a function named *isOlympic* that returns 1 if its array argument is an Olympic array, otherwise it returns 0.

If you are writing in Java or C#, the function signature is

`int isOlympic(int[] a)`

If you are writing in C or C++, the function signature is

`int isOlympic(int a[], int len)` where len is the number of elements in the array.

Hint: use a nested loop.

Copy and paste your answer here and click the "Submit answer" button

Submit answer to question 3

You should see a confirmation popup after hitting the submit button above.

If you have submitted all three answers individually, you can click complete test button below.

Before you press complete test button, make sure that you have submitted all three answers individually and received a confirmation popup saying your answers for that question was successfully submitted.

Once you click the complete test button, you can not modify or update your answers.

COMPLETE TEST