Master’s Program Sample Test

You will be asked to complete a real test as part of your application process.

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#. For each of the problems below, write the simplest, clearest solution you can, in the form of a short program.

SAMPLE TEST

1. **An array with an odd number of elements is said to be centered if all elements (except the middle one) are strictly greater than the value of the middle element. Note that only arrays with an odd number of elements have a middle element. Write a function that accepts an integer array and returns 1 if it is a centered array, otherwise it returns 0.**

Examples:

if the input array is	return
{1, 2, 3, 4, 5}	0 (the middle element 3 is not strictly less than all other elements)
{3, 2, 1, 4, 5}	1 (the middle element 1 is strictly less than all other elements)
{3, 2, 1, 4, 1}	0 (the middle element 1 is not strictly less than all other elements)
{1, 2, 3, 4}	0 (no middle element)
{}	0 (no middle element)
{10}	1 (the middle element 10 is strictly less than all other elements)

See correct answers to sample questions.

2. **Write a function that takes an array of integers as an argument and returns a value based on the sums of the even and odd numbers in the array.** Let X = the sum of the odd numbers in the array and let Y = the sum of the even numbers. The function should return X – Y

The signature of the function is:
int f(int[ ] a)

Examples

if input array is	return
{1}	1
{1, 2}	-1
{1, 2, 3}	2
{1, 2, 3, 4}	-2
{3, 3, 4, 4}	-2
{3, 2, 3, 4}	0
{4, 1, 2, 3}	-2
{1, 1}	2
{}	0

See correct answers to sample questions.

3. **Write a function that accepts a character array, a zero-based start position and a length. It should return a character array containing containing *length*characters starting with the *start*character of the input array.** The function should do error checking on the start position and the length and return null if the either value is not legal.

The function signature is:

char[ ] f(char[ ] a, int start, int len)

Examples

if input parameters are	return
{‘a’, ‘b’, ‘c’}, 0, 4	null
{‘a’, ‘b’, ‘c’}, 0, 3	{‘a’, ‘b’, ‘c’}
{‘a’, ‘b’, ‘c’}, 0, 2	{‘a’, ‘b’}
{‘a’, ‘b’, ‘c’}, 0, 1	{‘a’}
{‘a’, ‘b’, ‘c’}, 1, 3	null
{‘a’, ‘b’, ‘c’}, 1, 2	{‘b’, ‘c’}
{‘a’, ‘b’, ‘c’}, 1, 1	{‘b’}
{‘a’, ‘b’, ‘c’}, 2, 2	null
{‘a’, ‘b’, ‘c’}, 2, 1	{‘c’}
{‘a’, ‘b’, ‘c’}, 3, 1	null
{‘a’, ‘b’, ‘c’}, 1, 0	{}
{‘a’, ‘b’, ‘c’}, -1, 2	null
{‘a’, ‘b’, ‘c’}, -1, -2	null
{}, 0, 1	null

See correct answers to sample questions.

4. **Write a function to reverse an integer using numeric operators and without using any arrays or other data structures.**

   The signature of the function is:

   int f(int n)

Examples

if the input integer is	return
1234	4321
12005	50021
1	1
1000	1
0	0
-12345	-54321

See correct answers to sample questions.

5. **Write a function to return an array containing all elements common to two given arrays containing distinct positive integers. You should not use any inbuilt methods. You are allowed to use any number of arrays.**

   The signature of the function is:

   int[] f(int[] first, int[] second)

Examples

if input parameters are	return
{1, 8, 3, 2}, {4, 2, 6, 1}	{1, 2}
{1, 8, 3, 2, 6}, {2, 6, 1}	{2, 6, 1}
{1, 3, 7, 9}, {7, 1, 9, 3}	{1, 3, 7, 9}
{1, 2}, {3, 4}	{}
{}, {1, 2, 3}	{}
{1, 2}, {}	{}
{1, 2}, null	null
null, {}	null

null, null	null

See correct answers to sample questions.

6. **Consider an array A with n of positive integers. An integer idx is called a POE (point of equilibrium) of A, if A[0] + A[1] + … + A[idx – 1] is equal to A[idx + 1] + A[idx + 2] + … + A[n – 1]. Write a function to return POE of an array, if it exists and -1 otherwise.**

   The signature of the function is:

   int f(int[] a)

Examples

if input arrays are	**return**
{1, 8, 3, 7, 10, 2}	3 Reason: a[0] + a[1] + a[2] is equal to a[4] + a[5]
{1, 5, 3, 1, 1, 1, 1, 1, 1}	2 Reason: a[0] + a[1] is equal to a[3] + a[4] + a[5] + a[6] + a[7] + a[8]
{2, 1, 1, 1, 2, 1, 7}	5 Reason: a[0] + a[1] + a[2] + a[3] + a[4] is equal to a[6]
{1, 2, 3}	-1 Reason: No POE.
{3, 4, 5, 10}	-1 Reason: No POE.
{1, 2, 10, 3, 4}	-1 Reason: No POE.

See correct answers to sample questions.

Note: Please read the list of common programming errors that students have committed on our test.