

Honors Introduction to Computer Programming (COP 3223H) Exam #3 - C++

First Name: _____ **Last Name:** _____

1) (20 pts) Write a complete C++ program that determines whether or not two circles intersect. The user will enter in the x and y coordinates of the center of both circles as well as the radius of both circles. (Prompt them to enter each.) These values are guaranteed to be integers with an absolute value no more than 10,000. We consider two circles to intersect if at least one point from the first circle is either on the second circle or inside of the second circle. (Hint: Consider the distance between the centers of two circles that are tangent to one another. Also, you may use the math library, but there's a simple way to avoid doubles and square root and solve the problem accurately.)

2) (15 pts) A frog jumps around a circle with lily pads that have integer labels ranging from 0 to $n-1$. The labels and direction the frog jumps are clockwise. Each frog has a jumping distance j lily pads and starts at index 0. The frog stops jumping when it returns to lily pad 0. Write a complete C++ program that prompts the user to enter both n and j (positive integers no more than 1000000) and then prints out each lily pad the frog jumps on, including the initial 0 and ending 0. So, for example, if the user enters $n = 10$ and $j = 4$, your program should print out 0, 4, 8, 2, 6, 0. (No need to print the commas, just print each value followed by a space.)

3) (10 pts) Write a segment of C++ code that

- (a) declares an integer pointer `elevation` and a second integer, `n`.
- (b) reads in an integer into the variable `n` from the user.
- (c) dynamically allocates `elevation` to point to an array of size `n`.
- (d) reads in the following `n` integers from the user into the array allocated, in order.

4) (15 pts) A mountain sequence of `n` integers is a sequence, starting and ending with the same non-negative value, such that the first `k` pairs of integers are strictly increasing while the last `n-k-1` pairs of integers are strictly decreasing, where $k > 0$ and $n-k-1 > 0$. (The latter restriction prevents the sequence from being strictly increasing or strictly decreasing.) For example, the sequence 2, 6, 9, 12, 15, 2 is a mountain sequence, but the sequences 2, 5, 7, 7, 3, 2 and 3, 5, 9, 12, 15 are not. Complete the function below so that it returns true if the first `length` items of the array `elevation` form a mountain sequence, and false otherwise.

```
bool isMountain(int* elevation, int length) {
```

```
}
```

5) (10 pts) Complete the function below so that takes in an integer array elevation and an integer storing the length of the array, and returns the index that stores the peak of the mountain sequence stored in elevation. You may assume that the sequence passed to the function is a valid mountain sequence.

```
int peakLocation(int* elevation, int length) {
```

```
}
```

6) (12 pts) To scale a mountain specified by the array elevation, a climber must move from elevation[0] to elevation[1], then elevation[1] to elevation[2], and so forth. The minimum "skill" a climber needs to get to the peak is the largest "jump" between successive locations on the climb. For example, for the mountain sequence 2, 4, 6, 11, 13, 15, 4, 2, the minimum skill necessary to climb the mountain from the left is 5, since the third move the climber makes from index 2 to index 3 necessitates the climber moving from elevation 6 to elevation 11. (To climb from the right side, the minimum skill necessary is 11.) Complete the function below so that it returns the minimum skill necessary to climb a mountain from the left side. You may call the function peakLocation in your solution and you may assume the input sequence is a mountain sequence.

```
int minSkillFromLeft(int* elevation, int length) {
```

```
}
```

7) (13 pts) Write a similar function to number 6 that calculates the minimum skill necessary to climb the corresponding mountain described by the input sequence from the right.

```
int minSkillFromRight(int* elevation, int length) {
```

```
}
```

8) (3 pts) Write a single line of code to free the memory allocated to the elevation pointer from question 3.

9) (2 pts) Alfred Nobel invented dynamite. Unbeknownst to most of his family, he left most of his amassed wealth to be given away in the form of prizes for excellence in Peace, Literature, Physics, Chemistry, Medicine and Economics. What are these prizes known as today?

Scratch Page - Please clearly mark any work on this page you would like graded.