- You have approximately 80 minutes.

- This is a long exam, with many chances to demonstrate understanding. We encourage you to try the problems in the order that makes most sense to you, and to keep moving if you get stuck on one question.

- The exam is closed book, closed notes.

- Mark your answers ON THE EXAM ITSELF. If you are not sure of your answer you may wish to provide a *brief* explanation. All short answer sections can be successfully answered in a few sentences AT MOST.

- Make sure to write your answers clearly and legibly.

- Write your initials on the top right hand corner of each page.

First name	
Last name	
BMC/HC Email (What you use for gradescope)	

First and last name of student to your left	
First and last name of student to your right	

I agree to complete this exam without unauthorized assistance from any person, materials or device.

Signature:

For staff use only:

Total	/75

Q4. [40 pts] Programming

Complete the following four questions:

(a) [10 pts] Write a recursive method called `isPrime()`. Given a positive integer, determine if the number is prime or not. A prime number is a number that is only divisible by itself and 1.

Examples:

- `isPrime(3)−> true`
- `isPrime(15)−> false`
- `isPrime(2)−> true`
- `isPrime(1000)−> false`
- `isPrime(971)−> true`

If you cant come up with a recursive solution, you can use a loop for partial credit.

(b) [10 pts] Write a method `sumOfRange()` that takes in two numbers and returns the sum of all numbers between them (the numbers are inclusive, see the examples). Your method should use a loop.

Examples:

- `sumOfRange(1, 2)` $->$ 3
- `sumOfRange(10, 20)` $->$ 165
- `sumOfRange(20, 10)` $->$ 165
- `sumOfRange(-20, -10)` $->$ -165

(c) [10 pts] Write a method called `sumOf2Numbers()` that asks a user for two numbers, and then returns the sum of the two numbers. In this method, you cannot assume the user will always enter a number. If the user doesnt enter a number, prompt them again for a number until they input a number.

(d) [10 pts] Write a method `triangle()` that takes in a character and a number. The method should print out a right triangle where the number indicates the height of the triangle. The right triangle should consist of the character that is passed into the method.

Examples:

- `triange(*, 3) − >`

  ```
  *
  * *
  * * *
  ```

- `triangle(A, 5) − >`

  ```
  A
  A A
  A A A
  A A A A
  A A A A A
  ```