The University of Chicago
Department of Computer Science

CMSC 15200 – Introduction to Computer Science 2
Summer Quarter 2005
Homeword #1 (07/27/2005)
Due: 07/29/2005

Name:

Student ID:

Instructor: Borja Sotomayor

Do not write in this area

1	2	3	4	**TOTAL**

Maximum possible points: 45

Exercise 1 <<5 points>>

Write a program that will ask the user for three numbers and will then output the largest of the three.

```
Enter number #1: 42
Enter number #2: 23
Enter number #3: 37

The largest number is 42
```

Exercise 2 <<15 points>>

Improve the program from the previous exercise so that users can introduce as many numbers as they want. After introducing a number, the user will be asked if he wants to enter another number. If the user answers "Yes", then the user is asked for another number. If the user answers "No", the program will proceed to write the maximum, minimum, and average of all the numbers. Hint: This problem can be solved without having to store the individual numbers themselves anywhere. Hint #2: In C++, constants INT_MIN and INT_MAX represent the smallest possible integer and the largest possible integer, respectively.

```
Enter number #1: 42
Do you want to enter another number? Y

Enter number #2: 23
Do you want to enter another number? Y

Enter number #3: 37
Do you want to enter another number? N

The smallest number is 23, and the largest number is 42
The average is 34
```

For full credit (10 points, otherwise), when asking if the user wants to perform another conversion, check that the character is 'Y', 'y', 'N', or 'n' (without using an if-then statement). If the user introduces any other character, show an error message and ask him again whether he would like to perform another conversion.

Exercise 3 <<10 points>>

Write a program that finds the minimum number of bills and coins necessary to return a given amount of money (Note: we do not consider silver dollars, 50c coins, or bills bigger than $100). For example:

```
Enter amount: $3673.64

$100 bills: 36
$50 bills: 1
$10 bills: 2
$5 bills: 0
$1 bills: 3

25c coins: 2
10c coins: 1
5c coins: 0
1c coins: 4
```

Exercise 4 <<15 points>>

Write a program that, given an integer, determines if it is prime or not (if it is not prime, you do *not* need to print its factors).

Using a naïve algorithm (checking if the given number N is not divisible by all integers < N-1) will only earn you 10 points. For full credit, you will need to use an algorithm that doesn't require working throught *every single* integer < N-1. Hint: Don't try to come up with that algorithm by yourself. There is plenty written about primality tests on the web. Google and Wikipedia are your friends! Also, don't forget to cite your sources.

```
Enter a number: 15
15 is not prime.

Enter a number: 17
17 is prime.
```