



**The University of
Chicago**
Department of
Computer Science

CMSC 15200 – Introduction to Computer Science 2
Summer Quarter 2007
Homework #1 (07/25/2007)
Due: 07/27/2007 @ 5pm

Name:

Student ID:

Instructor:

Borja Sotomayor

Do not write in this area				
1	2	3	4	TOTAL

Maximum possible points: 45

NOTE: Remember the deadline refers to the submission of the homework code with hwsbmit. You can hand in this cover sheet, and any printed code, after the deadline (nonetheless, try to hand it in as soon as possible: the sooner you do, the sooner I can start grading the homework)



**The University of
Chicago**
Department of
Computer Science

CMSC 15200 – Introduction to Computer Science 2
Summer Quarter 2007
Homework #1 (07/25/2007)
Due: 07/27/2007 @ 5pm

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 enter another number, 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 enter another number.

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



**The University of
Chicago**
Department of
Computer Science

CMSC 15200 – Introduction to Computer Science 2
Summer Quarter 2007
Homework #1 (07/25/2007)
Due: 07/27/2007 @ 5pm

bigger than \$100). For example:

Enter amount: \$3673.64

\$100 bills: 36
\$50 bills: 1
\$20 bills: 1
\$10 bills: 0
\$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 two integers N and M, finds their *greatest common divisor*.

Using a naïve algorithm (trying *all* possible divisors for each of the two numbers, from 1 to N-1, and from 1 to M-1, and keeping track of what the largest common divisor is) will only earn you 10 points. For full credit, you will need to use an algorithm that doesn't require working through *every single* divisor. Hint: Don't try to come up with the algorithm yourself. There is a well-known classic algorithm to find the greatest common divisor of two numbers (this algorithm, in fact, is much easier to code than the naïve algorithm). Find out what that algorithm is and don't forget to cite your sources.

Enter number N: 15
Enter number M: 24
gcd is 3

Enter number N: 23
Enter number M: 15
gcd is 1