

Cs352 — Homework #2

Due Time: 9/16/03 (9:00PM)

Turnin ID: *cs352_assg2*

Turnin files: *Qremove.c*, *ExponentCheck.c*, *sum.c*

(PS: For example, you may turnin multiple files by UNIX command “turnin *cs352_assg1* yourfile1 yourfile2 yourfile3”. You may also turnin one file at a time by “turnin *cs352_assg1 file_you_want_to_turnin*”. Later turnin file will overwrite the old file which has the same filename. So if you turnin the same file multiple times, we only receive the last version you turned in. To see a list of the files you turned in, you may use the command “turnin -ls *cs352_assg1*”. For help information about turnin program, please use the command “turnin -h”. If you still have questions about turnin, please either stop by TA’s office hours or email TA’s.

Your code should follow the instructions in the “C coding guidelines”. In particular, pay attention to proper documentations)

We will put the sample executables in the class home dir at */home/cs352/fall03/sample_exec/assg2* on *lectura*. The purpose of these samples is to give you an idea how the expected solution look like (how it runs, how it gets input, how it format the output, etc). Please check your output format against the output generated by the sample executables with “diff” command. If you found any bugs with the sample executables or have questions on them, please contact us.

1. Create a C program called *Qremove* that reads a text from the standard input, and prints it to the standard output after all ‘Q’ letters have been removed. Note that the flow control commands in C are very similar to the ones in JAVA, so ‘while()’ so you should be familiar with (e.g.) the syntax to ‘while’ command.
2. It is well known that the function e^x can be computed by evaluating the the sum

$$\sum_{i=0}^{\infty} x^i / i! .$$

- (a) Write a function called *Exponent*, whose prototype is
float Exponent(float x, float eps)

that receives a float x and a float eps (stands for ε) and compute the sum

$$\sum_{i=0}^k x^i / i!$$

where k is the smallest integer such that $x^k/k!$ is no larger than eps . Pay attention to efficiency — try to minimize the number of arithmetic operations performed by the function.

- (b) Write a program called *ExponentCheck* that repeatedly asks the user to enter two numbers (x and eps), and prints the value returned by the function *Exponent*(x, eps) that you created for the previous question. The program should check that $0 < x < 1$ and $eps > 0$). The program should exit if the value 0 is entered to either x or eps . In this case the value 0 should be returned. If an illegal value is entered, the program should return 1.
3. Write a C program (named *sum.c*) that reads (by *scanf()*) a list of integers from the stdin and print (by *printf()*) to the stdout the number of integers, the minimum integer, the maximum integer and the sum of all the integers. The input integers are separated by spaces and/or newlines. A sample run of the program (where "sum" is the executable compiled from *sum.c*) is below:

```
lec:xxx > sum
1 2 3
4 5 6 7 8
8, 1, 8, 36
lec:xxx >
```

In the above example, 1 through 8 are the input numbers. "8, 1, 8, 36" are number of integers, min, max and sum respectively. You may assume the input are all valid integers of type "int". When there is no integer coming from stdin at all, your program should print "0, N/A, N/A, 0".

The macros *INT_MIN* and *INT_MAX* in *limits.h* containing the maximum and minimum possible value of an integer might help.