

# FALL 2004: CGS 2423 C FOR ENGINEERS

[PROGRAMMING ASSIGNMENTS 5 AND 6;

ASSIGNMENT 5 DUE NOVEMBER 24 IN CLASS. ASSIGNMENT 6 DUE NOVEMBER 29 IN CLASS.]

## Problem Description

You are required to design a calculator. This calculator reads commands from data files called `CommandsHW5.dat` and `CommandsHW6.dat`, which will be available from your class website. Your program should read the commands in sequential order, process them, perform the necessary calculations, and then print out the results in a neat, readable manner.

## Details

The first line of the file contains two characters giving the initials of the user (such as `GN`). The rest of the file consists of a sequence of commands. Each line of the command file contains one command. Each command starts with a character and may have zero or more operands. You may assume that the data file has no errors. The number of commands in the file is unknown. But your processing may stop as soon as the `Quit` command is processed. The commands are: *integer add*, *integer subtract*, *integer multiply*, *integer divide*, *change to uppercase*, *change to lowercase*, *separate*, *print out all digits*, *print out the k-th digit*, *round real number to desired number of decimal places*, *divide number into two*.

## Calculator Commands

- + i j**                    **[Integer Add]**  
Add integers `i` and `j` and print out result
- \* i j**                    **[Integer Multiply]**  
Multiply integers `i` and `j` and print out result
- i j**                    **[Integer Subtract ]**  
Subtract integer `j` from `i` and print out result
- / i j**                    **[Integer Divide ]**  
Divide integer `i` by `j` and print out result of integer division
- C Ch**                    **[Character Case Change ]**  
Change character `Ch` to uppercase and print it out
- c Ch**                    **[Character Case Change]**  
Change character `Ch` to lowercase and print it out
- P i k**                    **[Print k-th Digit ]**  
Print out the `k`-th digit of integer `i`
- R x i**                    **[Round Reals ]**  
Round double value `x` to `i` significant decimal places (see project 1 on p301 of your text for details)

- S x**                    **[Separate ]**  
Separate out the sign, integer part and fractional part of double value  $x$  (just like we did in class)
- D i x**                    **[Partition Integer ]**  
Given integers  $i$  and  $x$ , print out two integers  $j$  and  $k$ , where the sum of  $j$  and  $k$  equals  $i$ , and when you take  $x\%$  of  $i$  and truncate it you get  $j$
- H**                        **[Help ]**  
Print a short synopsis of all the available commands
- Q**                        **[Quit ]**  
Quit

## Additional Notes

For Assignment 5, you will only have to implement commands:  $+$ ,  $-$ ,  $*$ ,  $/$ , **H**, **Q**. For Assignment 6, you will need to implement the rest.

There is always a space between the command character and the operands (if any). For the division operation, the second operand will never be a 0. For the operation that changes to uppercase, you will always be given a character that is a lowercase letter of the alphabet (never a uppercase letter of the alphabet, a digit, a punctuation, or a special character). For the operation involving rounding a double value to the right decimal point, the output may have trailing 0's to pad it at the end. So the output for command "**R 3.567 1**" may be 3.6000. You will not be asked to round a negative number.

Print out a short synopsis of all the commands at the start of the program in order to guide the user. Make sure that your output echoes the commands you read in and provides the output in a readable format. Make sure you separate the output of one command from the next.

Start with 2 basic operators (**Add and Quit**), prepare a file with just these 2 commands and get your program tested and working for it. Then move on to other commands, one by one. If you have not gotten one of the commands to work, then make a dummy function that simply prints out a message saying "**Function Z not implemented yet!**"

## What to Submit

Submit a print out (hard copy) of your program and the output. As before, upload a "soft" copy of your program to the submission website at <ftp.cs.fiu.edu/pub/giri/incoming/cgs2423/jkerry001/> by logging in as **anonymous**. There will be no excuses accepted this time around. As before, if you are using Linux, then submit the **.c** file, and the **a.out** file. If you are running Windows and Visual C, then submit the source code, project file, the workspace file, the **.exe** file, and anything else that might be needed. Zip up all the files you want to submit into one zip file. Name the zipped file appropriately by using your login name and the assignment number (for example, **jkerry001\_Homework5.zip**). Make sure you put the zip file in the correct directory.

## Suggestions for the bored

**Extra Credit** Add extra commands related to having memory. Document your new commands properly. Print out any extra data files you use, and print out the corresponding output files.