Project [4]: Interactive Programs

Due Tuesday, 10/9/2018, 11:59 pm

Project Goals

The goals of this project are to:

1. Get students familiar with switch statements
2. Get students familiar with writing interactive programs
3. Get students familiar with character input/output functions

Important Notes:

1. **Formatting:** Make sure that you follow the precise recommendations for the output connate and formatting: for example, do not change the text in the first problem to differ from what is in the assignment. Your assignment will be auto-graded and any changes in formatting will result in a loss in the grade.
2. **Comments:** Header comments are required on all files and recommended for the rest of the program. Points will be deducted if no header comments are included.
3. **Restriction:** The use of `goto` statements anywhere within this program is prohibited. Points will be deducted if `goto` is used.

Problem 1

Write an interactive program that implements a simple calculator. The program should allow the user to choose a binary arithmetic operation and enter two terms to which to apply the operation. The program should then compute the result and display it to the user. Your calculator will have two modes: double-precision mode and integer mode. Double-precision mode will do calculations and output using variables of type `double`. Integer mode will do calculations and output using variables of type `int`. Your program should start in double-precision mode.

At the beginning, the program should output the following:

```
This program implements a calculator.
```

At every iteration, your program should ask the user for an option as follows:

```
Options:
1 - addition
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option:
```

Below is a description of what the program should do for each of the above options (items underlined are to be entered by the user):

1. Perform addition. For this option, the program should ask the user to enter two terms to be added, as follows:

   In double-precision mode:

   ```
   Enter first term: 1
   Enter second term: 1
   The sum is: 2.000000000000000
   ```

   In integer mode:

   ```
   Enter first term: 1
   Enter second term: 1
   The sum is: 2
   ```

2. Perform subtraction. For this operation, the program should ask the user to enter two terms to be subtracted, as follows:

   In double-precision mode:

   ```
   Enter first term: 4
   Enter second term: 2
   The difference is: 2.000000000000000
   ```

   In integer mode:

   ```
   Enter first term: 4
   Enter second term: 2
   The difference is: 2
   ```

3. Perform multiplication. For this operation, the program should ask the user to enter two terms to be multiplied, as follows:

   In double-precision mode:

   ```
   Enter first term: 2
   Enter second term: 3
   The product is: 6.000000000000000
   ```

   In integer mode:

   ```
   Enter first term: 2
   Enter second term: 3
   The product is: 6
   ```

4. Perform division. For this operation, the program should ask the user to enter two terms to be divided, as follows:

   In double-precision mode:

   ```
   Enter first term: 5
   Enter second term: 2
   The quotient is: 2.500000000000000
   ```

In integer mode:

```
Enter first term: 5
Enter second term: 2
The quotient is: 2
```

If the second term entered by the user is 0, the program should print:

```
Cannot divide by zero!
```

5. Toggle the type of calculator. By default, the program should perform its calculations using double-precision. However, if the user chooses this option, the program should change the type of calculator: either from `double` to `int` or vis versa.

   When switching from double-precision to integers the program should print the follow message:

   ```
   Calculator now works with integers.
   ```

   When switching from integers to double-precision the program should print the following message:

   ```
   Calculator now works with doubles.
   ```

6. Exit. The program should end.

Your program should function as follows (items underlined are to be entered by the user):

```
This program implements a calculator.
Options:
1 - addition
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option: 1
Enter first term: 1.525
Enter second term: 3
The sum is: 4.525000000000000
Options:
1 - addition
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option: 2
Enter first term: 4.873
Enter second term: 3.1
The difference is: 1.773000000000000
Options:
1 - addition
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option: 3
```

```
Enter first term: 4.56
Enter second term: 2.13
The product is: 9.712799999999998
Options:
1 - addition
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option: 4
Enter first term: 5.15
Enter second term: 9.8765
The quotient is: 0.521439781299043
Options:
1 - addition
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option: 4
Enter first term: 7.13
Enter second term: 0
Cannot divide by zero!
Options:
1 - addition
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option: 18
Invalid Option.
Options:
1 - addition
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option: 5
Calculator now works with integers.
Options:
1 - addition
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option: 1
Enter first term: 7
Enter second term: 9
The sum is: 16
Options:
1 - addition
```

```
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option: 4
Enter first term: 5
Enter second term: 2
The quotient is: 2
Options:
1 - addition
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option: 5
Calculator now works with doubles.
Options:
1 - addition
2 - subtraction
3 - multiplication
4 - division
5 - toggle calculator type
6 - exit program
Please enter your option: 6
```

Notes:

- If the user enters an invalid command, the program should print the menu again and ask for a valid option, as follows:
  ```
  Invalid Option.
  Options:
  1 – addition
  2 – subtraction
  3 – multiplication
  4 – division
  5 – toggle calculator type
  6 – exit program
  Please enter your option:
  ```
- For all options, except division, you can assume the user will give valid input
- When the calculator is in double-precision mode it should output 15 digits after the decimal point.

Save your program as `calc.c`

Challenge for problem 1 (10 extra credit points):
Turn the calculator into a scientific calculator. Change your calculator as follows:

```
        Options:
        1 – addition
        2 – subtraction
        3 – multiplication
        4 – division
        5 – exp(x)
        6 – log(x)
```

```
7 – toggle calculator type
8 – exit program
Please enter your option:
```

5. Compute `exp(x)`. For this option, the program should prompt the user to enter a single number as follows:

   In double-precision mode:

   ```
   Enter term: 4
   The result of exp(4.000000000000000) is: 54.598150033144236
   ```

   In integer mode:

   ```
   Cannot calculate with integers.
   ```

6. Compute `log(x)`. For this option, the program should prompt the user to enter a single number as follows:

   In double-precision mode:

   ```
   Enter term: 4
   The result of log(4.000000000000000) is: 1.386294361119891
   ```

   In integer mode:

   ```
   Cannot calculate with integers.
   ```

   This is the natural logarithm, it cannot be used with negative numbers. If the user enters a number that is less than or equal to zero, you program should print:

   ```
   Cannot take the log of that number!
   ```

Note:

- For this program you must use the `exp()` and `log()` functions from the math library. To do this:
  - Add `#include <math.h>` in your `calc_c.c` file
  - Add `-lm` to the compilation command: `gcc -o calc_c calc_c.c -lm`

Save your challenge separately as `calc_c.c`

Grading Rubric

Grading will be done for each problem as follows:

Correctly-named file	5%
Header comment	2%
Program compiles	5%
Correctly-reading data from terminal	28%
Correct result printed	60%

Submission details

To submit your project, you will have to use the submission script. You do this by either:

1. Working on an ECC machine
2. Working on the provided VMware
3. Secure Copying your files (See Mac Support for information)

To Submit your project:

- Have a directory called "project4"
- Save your *.c files in that directory
- To submit: **(don't type the '>' symbols)**

```
> cd project4
> submit
```

The submission script copies all files in the current directory to our directory. You may submit as many times as you like before the deadline, we only keep the last submission.

Academic Honesty

Academic dishonesty is against university as well as the system community standards. Academic dishonesty includes, but is not limited to, the following:

Plagiarism: defined as submitting the language, ideas, thoughts or work of another as one's own; or assisting in the act of plagiarism by allowing one's work to be used in this fashion.

Cheating: defined as (1) obtaining or providing unauthorized information during an examination through verbal, visual or unauthorized use of books, notes, text and other materials; (2) obtaining or providing information concerning all or part of an examination prior to that examination; (3) taking an examination for another student, or arranging for another person to take an exam in one's place; (4) altering or changing test answers after submittal for grading, grades after grades have been awarded, or other academic records once these are official.

Cheating, plagiarism or otherwise obtaining grades under false pretenses" constitute academic dishonesty according to the code of this university. Academic dishonesty will not be tolerated and penalties can include cancelling a student’s enrolment without a grade, giving an F for the course, or for the assignment. For more details, see the University of Nevada, Reno General Catalog.