

CS 241L - Data Organization

Spring 2022

Programming Assignment 2

Total points: 100

September 9th, 2022

In this programming assignment, you will print formatted strings using the `printf` function and work with numeric data types.

Part 0: Extracting files from tarball

The programming assignment is given to you as a compressed tarball. In order to extract the files from this tarball use the command

```
$ tar -xvf programming_assignment_2.tar.gz
```

This will create a folder with name 'Programming Assignment 2' in your current directory. The contents of this folder are the following items:

- 'Programming Assignment-2_F22.pdf'
- `madLib.c`
- `intPlay.c`

Part 1: Piano keys

On a piano, a key has a frequency, say f_0 . Each higher key (black or white) has a frequency of $f_0 * r^n$, where n is the distance (number of keys) from that key, and r is $2^{(1/12)}$.

Given an initial key frequency (an integer), output that frequency and the next higher key frequencies. Output each floating-point value with two digits after the decimal point, which can be achieved as follows:

```
printf("%.2lf", yourValue);
```

Ex: If the input is: 440.0 (which is the A key near the middle of a piano keyboard), the output is:

```
440.00 466.16 493.88 523.25 554.37
```

Note: Use one statement to compute $r = 2^{(1/12)}$ using the `pow` function (remember to include the math library). Then use that `r` in the subsequent statements that use the formula $f_n = f_0 * r^n$ with n being 1, 2, 3, and finally 4.

Instructions

1. Create a file named `piano_keys.c`.
2. Using the `scanf` function, read an integer from standard input as follows:

```
scanf("%d", &initialKey)
```


For the latter, don't forget to include the standard IO library with `#include <stdio.h>`.
3. Implement the necessary code compute and print the above sequence of frequencies from the given integer value read from standard input.
4. We will use this command to compile your code:

```
$ gcc piano_keys.c -Wall -Wuninitialized -Wconversion -o a.out -lm -std=c99
```

Part 2: Mad Libs

Mad Libs are activities that have a person provide various words, which are then used to complete short stories in unexpected (and hopefully funny) ways.

Instructions

1. Edit the file `madLib.c` provided to you to read the needed values from standard input, that the existing output statement(s) in `madLib.c` can use to output a short story.

Ex: If the input is:
`Eric Chipotle 12 cars`

The output is
`Eric went to Chipotle to buy 12 different types of cars.`
2. We will use this command to compile your code:

```
$ gcc madLib.c -Wall -Wuninitialized -Wconversion -o b.out -std=c99
```

Part 3: Playing with Integers

1. Compile the file `intPlay.c` provided to you in this assignment with the command:

```
$ gcc intPlay.c -std=c99 -o iPlay
```
2. Run `iPlay` and redirect its output to a file, so that you can save that output.

```
$ ./iPlay > intOutput
```

3. Create a file `intPlayAnswers` to place your answers to the four questions and submit the pdf version of this file. You could handwrite it and scan into a pdf or you may use your favorite text editor and convert it into pdf.

Each question refers to the `printf` it is associated with. You need to have the output handy so that you can refer to it when answering the questions.

What to submit:

Submit a tarball to Canvas with name `<Your-Student-ID>_2.tar` containing the following files:

Your file `piano_keys.c`

Your changes to `madLib.c`

Your file `intPlayAnswers.pdf`

`intOutput`

The command to archive these four files is

```
$ tar -cvf <Your-Student-ID>_2.tar
```

Grading Rubric:

If any of your C programs report errors or warnings using the respective commands stated the points given for the assignment will be zero. Otherwise the following rubric will be used:

+5 pt: Your submission consists of single tarball as specified in the ‘What to submit section.’

+5 pt: Your C files follows the class coding style.

+30 pt: The program `a.out` compiled from your file `piano_keys.c` passes different local `diff` tests.

+30 pt: The program `b.out` compiled from your file `madLib.c` passes different local `diff` tests.

+20 pt: Your answers provided in the file `intPlayAnswers.c` are correct.

+10 pt: The file `intOutput` passes a `diff` test.