

Lab #11: Program Layout and Command Line

Getting started

Download lab11 material from D2L

Enter Mimir IDE.

Change into the cse220 directory.

Create a new directory called lab11.

Change into the new directory.

Upload the lab material to Mimir IDE, save it in /home/(your_username)/cse220/lab11/

Implement the programs below in your lab11 directory.

Program 1: Command Line Arguments with File Redirection

Copy your code, **rot_command_line.c**, from lab09 into lab11. We will continue some work following that code.

For today's work, we want to use file redirection to write to and from files rather than using the command line. You should add an additional step to the program so that in addition to the command line parameters, it also reads a word and an integer from standard input, and writes the transformed word to standard output. You will then use file redirection when calling the program to read and write from files instead of the command line.

Compile your code as **rotCipher**. The program could be called as followed:

```
./rotCipher <input.txt >output.txt
```

Where input.txt has the following contents:

```
20 neilsen
```

There would be no command line output.

After execution, output.txt would have the following contents:

```
hcyfmyh
```

Program 2: Temperature Records

Complete the program **temperatures.c** to store temperatures (ints) recorded over a period of time. The program should store the temperatures in a global array variable. Assume that the maximum number of temperatures that may be stored is 500.

NOTE: All functions should be implemented with pointers. The only brackets present in your program should be for the array declaration. It is recommended that you implement functions one at a time. All averages should be rounded to one decimal point.

Add the following functions to your program:

- A function that finds the average of all temperatures stored.

- A function that finds the average of the last 5 recorded temperatures. If there are less than 5 temperatures recorded, return the average of all recorded temperatures.
- A function that records a new temperature. It must check if there is still space in the array.
- A function that prints the number of temperatures recorded.
- A function that prints all the temperatures recorded.

Organize the program so the function prototypes are presented first, and the definitions after the main function.

The main function is not allowed to have any scanf or printf function calls.

Your program should ask the user to select an option from the following, and repeatedly ask them to choose an option until option (F) is selected:

- (A) Compute and print average of all temperatures,
- (B) Compute and print the average of the last five temperatures added,
- (C) Add a new temperature to the array,
- (D) Print the number of temperatures recorded,
- (E) Print all the temperatures recorded, or
- (F) Exit the program.

Compile and test your program.

See example inputs/outputs on the next page:

Suppose I have already inputted 4 temperatures: [20, 21, 22, 23]

Now I want to add 24, and try all the options:

```
Pick one of the following options:
1: Add a new temperature to the array.
2: Print the number of temperatures recorded.
3: Print all the temperatures recorded.
4: Compute and print average of all temperatures.
5: Compute and print the average of the last five temperatures added.
6: Exit the program.
1
You chose option 1.
What is the new temperature?
24
You added the temperature 24.

Pick one of the following options:
1: Add a new temperature to the array.
2: Print the number of temperatures recorded.
3: Print all the temperatures recorded.
4: Compute and print average of all temperatures.
5: Compute and print the average of the last five temperatures added.
6: Exit the program.
2
You chose option 2.
There are 5 temperatures recorded.

Pick one of the following options:
1: Add a new temperature to the array.
2: Print the number of temperatures recorded.
3: Print all the temperatures recorded.
4: Compute and print average of all temperatures.
5: Compute and print the average of the last five temperatures added.
6: Exit the program.
3
You chose option 3.
Temperatures recorded are: [20, 21, 22, 23, 24, ]
```

```
Pick one of the following options:
1: Add a new temperature to the array.
2: Print the number of temperatures recorded.
3: Print all the temperatures recorded.
4: Compute and print average of all temperatures.
5: Compute and print the average of the last five temperatures added.
6: Exit the program.
4
You chose option 4.
Total Average is 22.0.

Pick one of the following options:
1: Add a new temperature to the array.
2: Print the number of temperatures recorded.
3: Print all the temperatures recorded.
4: Compute and print average of all temperatures.
5: Compute and print the average of the last five temperatures added.
6: Exit the program.
5
You chose option 5.
Last 5 Average is 22.0.

Pick one of the following options:
1: Add a new temperature to the array.
2: Print the number of temperatures recorded.
3: Print all the temperatures recorded.
4: Compute and print average of all temperatures.
5: Compute and print the average of the last five temperatures added.
6: Exit the program.
6
user@mimir: ~/cse220/lab10 >
```