

Programming Assignment #1

Lab Rats

CS 2308.256 Spring 2017

Instructor: Jill Seaman

Due: Tuesday, 1/31/2017: upload electronic copy by 9:00am.

Write a program that stores the following data about a lab rat in a structure called Rat:

- The rat's name
- The rat's weight in ounces
- The amount of time in seconds it took the rat to run the maze

The program should keep an array of 9 of these structures. Each element is for a different rat in the lab. When the program runs, it should initialize the array using the following data:

Ben, 6.5, 29
Fred, 5.8, 31
Splinter, 7.7, 33
Lima Bean, 6.5, 27
Black Bean, 5.8, 31
Pinto Bean, 7.7, 33
Skat, 6.5, 29
Big Pook, 9.8, 34
Tiger, 7.7, 33

It should then display a table that lists each rat's name, weight, and time. The program should also calculate and display the average weight of all of the rats. The names and the times of the rat who ran the maze the fastest and the rat who ran the maze the slowest should also be displayed.

Your program must include two functions, one to calculate the average weight and one to find the fastest and slowest rats. These functions should take an array of Rats as an argument. The first one should return the average weight, the second one should use reference parameters to return the index of the fastest and slowest rats to the caller.

Output:

- The data does not need to line up in columns, but it should be easy to read.
- The average weight should be formatted to one decimal place.
(Hint: `cout<<fixed<<setprecision(1)`)

Sample output:

```
Ben 6.5 29
Fred 5.8 31
Splinter 7.7 33
Lima Bean 6.5 27
Black Bean 5.8 31
Pinto Bean 7.7 33
Skat 6.5 29
Big Pook 9.8 34
Tiger 7.7 33

Average weight: 7.1
Fastest Rat: Lima Bean Time: 27
Slowest Rat: Big Pook Time: 34
```

Additional Requirements:

- Your program **must compile** and run, otherwise you will receive a score of 0.
- Your program must give correct results for any rat data. The grader will change the names, weights, and times in your program, and it should still give correct results.
- Please do not use any features of C++ that we have not yet covered in class (use features from Chapters 1-7, 11 only). Do not use classes!
- **Style:** See the Style Guidelines document on the course website. Especially pay attention to the **comments** required for the top of the file and for each function. The grader will deduct points if your program violates the style guidelines.

Logistics:

Name your file **assign1_XXXXX.cpp** where XXXXX is your TX State NetID (your txstate.edu email id).

There are two steps to the turn-in process:

1. Submit an electronic copy using the Assignments tool on the TRACS website for this class.
2. Submit a printout of the source file at the beginning of class, the day after the assignment is due. Please **print your name on top of the front page**, and staple if there is more than one page.

See the assignment policy on the course website (cs.txstate.edu/~js236/cs2308) for more details.