Middle East Technical University | Department of Computer Engineering

CENG 140

C Programming

Fall' 2020-2021
Take-Home Exam 2

Emre Külah
kulah@ceng.metu.edu.tr
Due date: Dec 27, 2020, Sunday, 23:59

MASTER CENG

1 Overview

Our department decided to organize a challenge called MasterCeng. Students will develop 5 different projects and submit their codes in a single submission. A jury formed from our instructors will evaluate each project and give them a score. Each submission will be evaluated as sum of its projects scores. Average score of the challenge and the winner will be announced.

The jury will score the projects according to programming language used to develop the project. The instructors have weights for each programming languages:

Instructor	**C**	**C++**	**Python**	**Java**
Instructor-1	2.4	2.6	4.2	4.0
Instructor-2	4.4	4.3	2.2	3.0
⋮	⋮	⋮	⋮	⋮
Instructor-n-1	3.8	5.0	1.2	1.3
Instructor-n	3.3	3.7	2.9	4.5

Jury will get the submission, the programming languages that projects written in, and will score each project according to programming languages.

	Project-1	**Project-2**	**Project-3**	**Project-4**	**Project-5**
Submission	C	Python	Python	Java	C++

Before evaluating the submission, the jury members have to be decided. Let's say size of the jury is 3 and Instructor-1, Instructor-2 and Instructor-n are selected.

```
Jury-1 Weights (Instructor-1) => 2.4 2.6 4.2 4.0
Jury-2 Weights (Instructor-2) => 4.4 4.3 2.2 3.0
Jury-3 Weights (Instructor-n) => 3.3 3.7 2.9 4.5
```

So calculation will be:

```
Project-1: Jury-1 Weight + Jury-2 Weight + Jury-3 Weight => (2.4+4.4+3.3) (written in C)
Project-2: Jury-1 Weight + Jury-2 Weight + Jury-3 Weight => (4.2+2.2+2.9) (written in Python)
Project-3: Jury-1 Weight + Jury-2 Weight + Jury-3 Weight => (4.2+2.2+2.9) (written in Python)
Project-4: Jury-1 Weight + Jury-2 Weight + Jury-3 Weight => (4.0+3.0+4.5) (written in Java)
Project-5: Jury-1 Weight + Jury-2 Weight + Jury-3 Weight => (2.6+4.3+3.7) (written in C++)
```

So final score for the submission will be:

```
(2.4+4.4+3.3) + (4.2+2.2+2.9) + (4.2+2.2+2.9) + (4.0+3.0+4.5) + (2.6+4.3+3.7) = 50.8
```

Different jury members will be selected for each submission's evaluation. At the end, average score for the challenge and the winner will be announced.

2 Tasks

- **float** create_instructors(int* size):** function gets instructor info from command line. It is going to read from stdin and create a list of instructors with their programming language weights. First integer input will be the number of instructors, following each line with four input floats will define an instructor. During this process, it sets size argument to keep size of the instructors list in main function.
  Input format:

```
<number of instructors>
<C_weight> <C++_weight> <Python weight> <Java weight>
<C_weight> <C++_weight> <Python weight> <Java weight>
<C_weight> <C++_weight> <Python weight> <Java weight>
...
<C_weight> <C++_weight> <Python weight> <Java weight>
```

- **int** create_submissions(int* size):** function gets submission info from command line. It is going to read from stdin and create a list of submissions with corresponding programming language that is written. First integer input will be the number of submission, following each line with five input integers will define a submission. During this process, it sets size argument to keep size of the submissions list in main function.
  Input format:

```
<number of submissions>
<PL of Project-1> <PL of Project-2> <PL of Project-3> <PL of Project-4> <PL of Project-5>
<PL of Project-1> <PL of Project-2> <PL of Project-3> <PL of Project-4> <PL of Project-5>
<PL of Project-1> <PL of Project-2> <PL of Project-3> <PL of Project-4> <PL of Project-5>
...
<PL of Project-1> <PL of Project-2> <PL of Project-3> <PL of Project-4> <PL of Project-5>
```

  Programming languages are enumerated as follows:

```
C: 0
C++: 1
Python: 2
Java: 3
```

- **int* create_jury():** function gets a number of integers. It is going to read from stdin and create the jury members. First integer input will be the size of the jury team, rest of the integers will be the index of the instructors.
  Input format:

```
<size of jury> <index_1> <index_2> ... <index_n>
```

- **float* calculate_submission_scores(int **submissions, int size, float **instructors):** function takes submissions and instructors as argument. It is going to call **create_jury** function and get jury members for each submission. It is going to calculate scores of the submissions and going to save them into a list.
- **float find_average_score(float *scores, int size):** function takes scores as argument. It is going to calculate average score for challenge, which is the average score computed overall submissions.
- **print_winner(int** submissions, int size, float* scores):** function takes submissions and scores as argument. It is going to find winner and it is going to print detail of the submission.
  Example print:
  Let's say winner submission is: C - Python - Python - Java - C++

```
0 2 2 3 1
```

- **void print_double_array(void **array, int type, int size):** function takes a 2D array, size of the array and type of the array as argument. If type value is zero then the input array will be instructors. If type value is one then the input array will be submissions. It is going to print the given array's content.

3 Example

```
int main(){
    ...

    /* get instructors */
    instructors = create_instructors(...);
    printf("Instructors:\n");
    print_double_array(...instructors as input...);

    /* get submissions */
    submissions = create_submissions(...);
    printf("Submissions:\n");
    print_double_array(...submissions as input...);

    /* calculate scores */
    scores = calculate_submission_scores(...);

    /* print scores */
    for (i = 0; i < #size_of_scores#; i++)
    {
        printf("Submission-%d score: %.2f\n", ...);
    }

    /* calculate average */
    avg = find_average_score(...);
    printf("Average: %.2f\n", avg);

    /* find and print winner */
    print_winner(...);

    return 0;
}
```

Input - Definition

```
Ins-1: 2.2 2.3 2.4 2.5
Ins-2: 3.0 3.1 3.2 3.3
Ins-3: 1.1 1.2 1.3 1.4
Ins-4: 1.7 1.8 1.9 2.0
Ins-5: 3.9 4.0 4.1 4.2

3 Submissions:
C++ - Python - Java - C++ - C
Python - C++ - C++ - Python - C
Java - Python - C++ - C - C

Jury Members (Size of 3):
Submission1: Ins-3, Ins-2, Ins-1
Submission2: Ins-3, Ins-4, Ins-5
Submission3: Ins-1, Ins-2, Ins-5
```

Input in stdin:

```
5
2.2 2.3 2.4 2.5
3.0 3.1 3.2 3.3
1.1 1.2 1.3 1.4
1.7 1.8 1.9 2.0
3.9 4.0 4.1 4.2
3
1 2 3 1 0
2 1 1 2 0
3 2 1 0 0
3 2 1 0
3 2 3 4
3 0 1 4
```

Output

```
Instructors:
2.20 2.30 2.40 2.50
3.00 3.10 3.20 3.30
1.10 1.20 1.30 1.40
1.70 1.80 1.90 2.00
3.90 4.00 4.10 4.20
Submissions:
1 2 3 1 0
2 1 1 2 0
3 2 1 0 0
Submission-1 score: 33.60
Submission-2 score: 35.30
Submission-3 score: 47.30
Average: 38.73
Winner: 3 2 1 0 0
```

4 Specifications

- If two or more of the submissions get same score, the winner of the challenge will be the one with the minimum index.
- In print_double_array function, there will be **NO** extra whitespace at the end of the lines and there will be extra newline character at the end.
- In print_winner function, there will be **NO** extra whitespace at the end of the lines and there will be extra newline character at the end.
- Floating points will be printed with 2 digit precision.

5 Regulations

- **Programming Language:** C
- **Libraries and Language Elements:**
  You should not use any library other than *"stdio.h"*, *"stdlib.h"*. You can use conditional clauses (switch/if/else if/else), loops (for/while), allocation methods (malloc, calloc, realloc). **You can NOT use any further elements beyond that (this is for students who repeat the course).** You can define your own helper functions.
- **Compiling and running:**
  DO NOT FORGET! YOU WILL USE ANSI-C STANDARTS. You should be able to compile your codes and run your program with given **Makefile**:

  ```
  >_ make the2
  >_ ./the2
  ```

  If you are working with ineks or you are working on Ubuntu OS, you can feed your program with input files instead of typing inputs. This gives the input from stdin and an equivalent of typing inputs. This way you can test your inputs faster:

  ```
  >_ ./the2 < inp1.txt
  >_ ./the2 < inp2.txt
  ```

- **Submission:**
  You will use CengClass system for the homework just like Lab Exams. You can use the system as an editor or work locally and upload the source files. Late submission IS NOT allowed, it is not possible to extend the deadline and **please do not ask for any deadline extensions**.
- **Evaluation:** Your codes will be evaluated based on several input files including, but not limited to the test cases given to you as an example. You can check your grade with sample test cases via CengClass system but do not forget it is not your final grade. Your output must give the exact output of the expected outputs. It is your responsibility to check the correctness of the output with the invisible characters. Otherwise, you can not get a grade from that case. If your program gives correct outputs for all cases, you will get 100 points.
- **Cheating: We have zero-tolerance policy for cheating**. People involved in cheating will be punished according to the university regulations and will get 0. Sharing code between each other or using third party code is strictly forbidden. Even if you take a "part" of the code from somewhere/somebody else - this is also cheating. Please be aware that there are "very advanced tools" that detect if two codes are similar. So please do not think you can get away with by changing a code obtained from another source.