

CSE 102 Programming Assignment 1

DUE

October 15, 2024, 23:55

Description

- This is an individual assignment. Please do not collaborate
- If you think that this document does not clearly describe the assignment, ask questions before its too late.

This is a C Programming assignment. You will write a C program according to the following description.

- Your program captures user input, learns and tests a simple classifier. This assignment is about using control statements, performing arithmetic operations and simple input/output.
- You are expected to divide the solution into blocks and use functions.

Program Description

There are three clusters(groups) of points in a 2 dimensional cartesian coordinate system. Every point has an x and y coordinate.

Program Flow

- The program first accepts center points for three clusters (Cluster 1, 2 and 3).
- After reading center points of three clusters, the program continues to capture point coordinates.
- Once a point coordinate is captured, the program decides to put this point to a particular cluster.
- After including a point in one of the three clusters, the program updates the center point of the chosen cluster.
- The program stops after capturing N number of points. (N is a macro)
- At the end of the program, averages of x,y coordinates of every cluster and the total number of points included in each cluster is printed.
- Format:

```
Cluster 1: x_coordinate_of_the_center, y_coordinate_of_the_center, number_of_points_in_cluster_1
Cluster 2: x_coordinate_of_the_center, y_coordinate_of_the_center, number_of_points_in_cluster_2
Cluster 3: x_coordinate_of_the_center, y_coordinate_of_the_center, number_of_points_in_cluster_3
```

How to decide which cluster a point belongs to?

- Given a point, calculate distances of this point to three cluster centers.
- Choose the closest cluster if there is at least X percent difference between the closest one and the next closest one. (X is a macro)
- If there is less than X percent difference, discard the point. Ignore the point and don't change any center points of any clusters.
- If the point was discarded, print the following message:

```
Point x_coordinate, y_coordinate was discarded.
```

- If a point was included in one of the clusters(Example: Cluster 1), print the following message:

```
Point x_coordinate, y_coordinate was included in Cluster 1
```

How to update the center point of a clusters?

- Update x and y coordinates of the center point of the cluster by calculating a new average x and average y by including the new point. You have to include the total number of points in the calculation.
- $\text{new_x} = (\text{old_x} * \text{number_of_points} + \text{point_x}) / (\text{number_of_points} + 1)$

Input Format

```
x_coordinate y_coordinate
```

Remarks

- Assume error-free inputs.
- Be careful with **divide-by-zero** situations. You can avoid them by perturbing the 0 results with very small numbers. (if the divider is 0, make it `0 + EPSILON`. Define a macro `EPSILON` and make it a small floating point number. (ex: 0.001))
- **Do not print anything other than the expected output.**
- You cannot use pointers and other things which are not covered in class.
- Using input/output redirection is advised. (Do not submit any of the files you used for testing).
- Test your program. Hand-trace the execution.

Turn in:

- Source code of a complete C program. Name of the file should be in this format: `<full_name>_PA1.c`.
- Example: `john_von_neumann_PA1.c`. Please do not use any Turkish special characters.
- You don't need to use an IDE for this assignment. Your code will be compiled and run in a command window.
- Your code will be compiled and tested on a Linux machine(Ubuntu). GCC will be used.
- Make sure you don't get compile errors when you issue this command : `gcc <full_name>_PA1.c`.
- A script will be used in order to check the correctness of your results. So, be careful not to violate the expected output format.
- Provide comments unless you are not interested in partial credit. (If I cannot easily understand your design, you may loose points.)
- You may not get full credit if your implementation contradicts with the statements in this document.

Late Submission

- Late submission is **NOT** accepted.

Grading (Tentative)

- Max Grade : 100.

All of the followings are possible deductions from Max Grade.

- No submission: -100.
- No demo: -100
- Unable to explain the code: -100
- Compile errors: -100.
- Irrelevant code: -100.
- Major parts are missing: -100.
- Unnecessarily long code: -30.
- Using language elements and libraries which are not allowed: -100.
- Not caring about the structure and efficiency: -30. (avoid using hard-coded values).
- Significant number of compiler warnings: -10.
- Not commented enough: -5. (Comments are in English)
- Source code encoding is not UTF-8 and characters are not properly displayed: -5. (You can use 'Visual Studio Code', 'Sublime Text', 'Atom' etc... Check the character encoding of your text editor and set it to UTF-8)
- Fails at properly catching user input: -20.
- Fails during cluster update. (wrong calculation): -30.
- Fails during cluster update. (program crashes): -70.
- Fails to print correct output (wrong output): -30.
- Fails to print correct output (program crashes): -70.
- Infinite loop: -90.
- Prints anything other than the expected: -30.
- Submission includes files other than the expected: -10.
- Submission does not follow the file naming convention: -10.
- Sharing or inheriting code: -200.

Note: Some of these items are not independent. So, you cannot expect isolation of many of them. For example, if you cannot catch user input correctly, your tests will fail. Partial grading is not guaranteed.