PA0: C Review (Cat Beauty Competition 🐱)

[1]

[1] Image generated by Google Gemini

Overview

In this assignment, you will review some C programming concepts using a fun pet cat theme 🐱. You will practice working with ***structures, functions, pointers, and standard input/output***. You will also learn how to receive inputs from a file using the **command line** and generate output of your code to an output file, and then compare the results. This process will help you to compare your code's output with large inputs and outputs.

Important Restrictions:

- Do NOT use dynamic memory allocation for this PA0.
- Do NOT use file I/O inside your code (anywhere. A line of code with fprintf/fscanf will result in zero in the whole assignment.
- Use only scanf and printf for input and output.
- Input will be redirected from a file using ***command-line*** redirection.
- Output will be redirected to an output file using command-line redirection.

Problem Description: Cat Beauty Competition 🐱

UCF students, faculty, and staff are organizing a Cat Beauty Competition to find the cutest cat on campus!

Each participating cat has:

- A unique Cat ID (int)
- A Cat Name (single word, all upper case, max length 10)
- Beauty scores from three judges (int)

Your program must:

1. Read the Cat ID, Cat Name, and three beauty scores for N cats (N ≤ 500).
2. Display each cat's:
   – Cat ID
   – Cat Name
   – Three beauty scores (int)
   – Average beauty score (rounded to 2 decimal places)
3. Determine and display:
   – the cat Name of the cutest cat based on the average beauty score
4. Count and calculate the percentage (up to two decimal places) of how many cat names start with each letter of the English alphabet (A–Z).
   – If no cat name starts with a letter, display 0 for that letter.

All output must be printed using standard output.

Specific Requirements:

- You must use an array of structures.
- The structure name must be cat (do NOT change the name).
    - Do NOT use typedef.
    - The structure must contain exactly the following fields :
      – cat_ID (int) → Cat ID
      – name, a character array for Cat Name
      – other fields as needed (e.g., beauty scores)
      - average (float)
- **Required Functions:**
    - All input must be read using the readData function
      void readData(struct cat *cats, int *c);
        - This function:
            - Receives an initial empty cats array of max size and a reference to an int
            - It reads data from standard input
            - Fills the structure array passed to the function
            - Update c with the actual cat count in the input
            - Computes each cat's average beauty score
            - Must NOT find the maximum average

    - To find the cutest cat, write the following function:

      int Cat getCutestCat(struct cat *s, int n);

      This function:
      • Takes an array of structures and its size
      • Returns the index that contains the cutest cat based on the highest average beauty score
      • Must be called from main() after readData()

Input Format (Standard Input)

• First line: Integer N (number of cats, N ≤ 500)
• Next N lines:
 – Cat ID
 – Cat Name
 – Three integers representing beauty scores

Sample Input:

```
6
861022 CAESAR 65 72 56
851102 CLEO 78 45 80
860501 PETRA 55 75 90
841205 HANA 75 80 95
850630 CUDDLE 40 50 48
460734 COCO 65 73 85
```

Output Format (Standard Output)

For each cat, print:

CatID CatName score1 score2 score3 average

Then print:

Cutest cat is <Cat Name>

Then print the alphabetical count and percentage (up to two decimal places) of cat names based on the requirement (first letter of the name):

```
A: x p1%
B: y  p2%
...
Z: z p26%
```

Every letter from A to Z must be printed, even if the count is zero.

Sample Output:

```
861022 CAESAR 65 72 56 64.33
851102 CLEO 78 45 80 67.67
860501 PETRA 55 75 90 73.33
841205 HANA 75 80 95 83.33
850630 CUDDLE 40 50 48 46.00
460734 COCO 65 73 85 74.33
Cutest cat is HANA
A: 0 0.00%
B: 0 0.00%
C: 4 66.67%
D: 0 0.00%
E: 0 0.00%
F: 0 0.00%
G: 0 0.00%
H: 1 16.67%
I: 0 0.00%
J: 0 0.00%
K: 0 0.00%
L: 0 0.00%
M: 0 0.00%
N: 0 0.00%
O: 0 0.00%
P: 1 16.67%
Q: 0 0.00%
R: 0 0.00%
S: 0 0.00%
T: 0 0.00%
U: 0 0.00%
V: 0 0.00%
W: 0 0.00%
X: 0 0.00%
Y: 0 0.00%
Z: 0 0.00%
```

Testing your code using command line

The following list of commands will be very useful for your programing assignments as you will have to deal with large input and outputs. Entering all the inputs will be time consuming. So, using the commands to pass the data from an input file will make your testing much easier.

Compiling in testing in command line (You can use replit and use the shell that you will find just beside the Console):

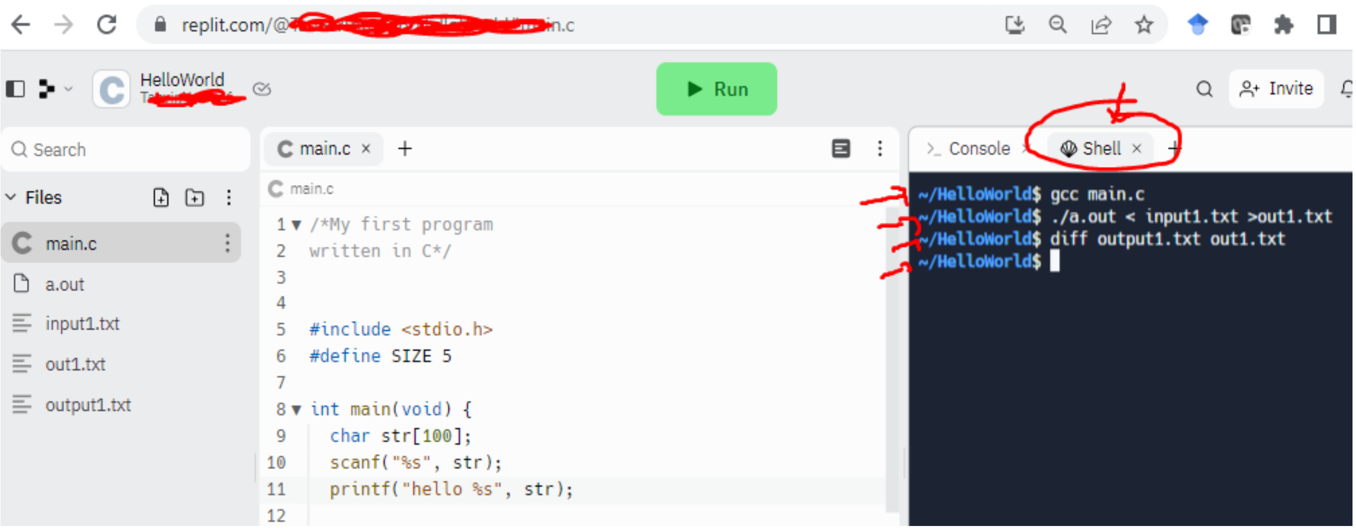
Use the following command for testing. (Don't put the $ sign. It is just an indicator that is a command)

$gcc CatContest.c //this will compile your c file and generate a.out file as an executable file if your code compiles successfully

$./a.out < sample_in.txt >out.txt //if there is no error, run your code by this command. It will take input from sample_in.txt file and the output will be generated to out.txt file

You can use various commands to compare your output file against the given sample output file. Note that in your programming assignments you have to exactly match the output with the sample output to get credit.

$diff out.txt sample_out.txt //the result will be blank if there is no difference between the files
$ diff -c out.txt sample_out.txt //this command will show ! symbol to the unmatched lines.
$ diff -y out.txt sample_out.txt //very useful to compare the files side by side to see where exactly it is not matching



```
/*My first program
written in C*/


#include <stdio.h>
#define SIZE 5

int main(void) {
  char str[100];
  scanf("%s", str);
  printf("hello %s", str);

```

```
~/HelloWorld$ gcc main.c
~/HelloWorld$ ./a.out < input1.txt >out1.txt
~/HelloWorld$ diff output1.txt out1.txt
~/HelloWorld$
```

Submission:

The purpose of this submission is to make yourself familiar with the Eustis system.

Please follow the coding style guide provided in the Webcourses. Solve and test your code in Eustis.

You must submit your files in Webcourses under the PA0 assignment.

You need to submit three files:

- **CatContest.c**
- **Cat.h**
- **commands.jpg (**1 screenshot should show that you have used the command line shown above to compile and execute your code **in Eustis** and the diff says no difference)

The Cat.h file should contain the following:

```
#define MAXSTRLEN 10

#define MAXCATS 500



struct cat {
    int cat_ID;           // Cat ID
    char name[MAXSTRLEN+1];           // Cat name (single word)
    int score1;
    int score2;
    int score3;
    float average;
};


void readData(struct cat *cats, int *c);

int getCutestCat(struct cat *s, int n);
```

```
Please include Cat.h file to your CatContest.c file.
#include "Cat.h" //put this line in your CatContest.c code right after including the
header files.

//after inlucindg Cat.h file to your CatContest.c, you don’t need to redefine the struct. Just
write all the functions and main function
```