ACS130 Introduction to Systems Engineering and Software
C Programming (ACS130-002)

Tara Baldacchino (t.baldacchino@sheffield.ac.uk, Room D13, AJB)

Assignment weighting: 6% of module mark

Assignment released: Friday 15 November 2019 (Semester 1, Week 7)

Assignment due: Midnight Monday 25 November 2019 (Semester 1, Week 9).

How to submit

The assignment must be submitted to Blackboard (MOLE) by **Midnight Sunday 25 November 2019**, at the link **ACS130 > Assessment > C Programming ACS130-002.** You must upload you *.c file. There is a separate drop box for ACSE and Aero students so please submit to the correct drop box. Mac users please make sure you submit your .c file and nothing else. **If, when I download your program from Blackboard (MOLE) I cannot run the file because it is not a .c file (or for any other reason), you will lose 1 mark.**

How the marking will be done

The assignment will be marked by appointment during the timetabled ACS130 lab session on Tuesday 26 Nov (Aero) and Wednesday 27 & Friday 29 Nov (ACSE) (Week 9). **You will be given an appointment time before the timetabled lab session.** The marker will be a graduate teaching assistant (GTA), and the module leader will be moderating the process. The marker will mark the assessment in the lab and will give the overall mark, individual component marks and comments on performance on the program. The attached mark sheet provides a guide to the areas on which feedback will be provided. The electronic submission will be reviewed after the lab session, by the module leader and marks may be moderated.

How you should work

This is an **individual assignment**, and it must be wholly your own work. You should not discuss a solution to this assignment with anyone else, and you should not work with anyone else to produce a solution.

You must complete this assignment in your own time, **NOT** in ACS130 lab sessions. You may work on the University computers, or on your own computer. **If you work on your own laptop, you are advised to test your program on the managed desktops.** You should be aware that a C program which works in another development environment may not work in the CodeBlocks environment on the University computers.

Learning outcomes

In doing this assignment, you will demonstrate your ability to program in C to manipulate files, output to the screen, implement the selection and repetition constructs, and manipulate variables, as well as use functions.

Assignment briefing

Write a C program to:

- Read characters (no spaces but a combination of numbers, letters and symbols) all on the same line from the keyboard **and at the same time** write the characters to the file File1.txt (all on the same line), then close the file.
- Read the characters from the file "File1.txt" that you have just created. (Hint: read the characters one at a time, using fscanf or fgetc), at the same time printing only the number characters to the screen, printing the count of how many numbers there are and also printing the sum of these numbers.
- You need to program defensively and check to see if the file exists.
- In a function `float mean(float sum, float N)` where `sum` is the sum of the numbers entered, and `N` is the total number of numbers.
    - calculate the mean of the numbers
    - return this mean to main
- Print to the screen the mean (to 2 d.p.) of the numbers from main.

For example, your program should look like this when input sf98375hf^%839*&% is used:

```
please input chars on a line, no spaces:
sf98375hf^%839*&%
Numbers found in the line:
9
8
3
7
5
8
3
9
sum is 52.0
no of numbers is 8.0
The mean is 6.50
```

Please note:

1. The only library you may include is stdio.h, therefore you will not be allowed to use pre-defined functions from any other library. **You will get 0 if you use more libraries.**
2. You may not use arrays or stings (which will be covered in semester 2). **You will get 0 if you use arrays or strings.**
3. You may not use global variables. These are variables declared outside main. **You will get 0 if you use global variables.**
4. When your program is marked, the markers will type in a range of characters to test your program. You should test your program thoroughly before you submit it.
5. Your program must have meaningful comments, including header comment, and be well-laid out and readable.
6. **Helpful tip 1:** the end of line character is the newline, so you can compare the input string to `'\n'`.
7. **Helpful tip 2:** You need to think about how you're going to convert the numerical characters into numbers in order to be able to find the sum. Use the ASCI table (see www.ascii-code.com) and refer to details given verbally in the lecture. You are not allowed to use the function `int atoi()` because this is in the stdlib.h library.

Marking Scheme for Software

Marking Criterion	Mark
Does the code use any other library other than stdio.h or uses arrays/strings or uses global variables- 0 marks!	
Looking at the code Program layout and readability including: is the code indented, does it have comments, header comments and meaning variable names? (0.25 mark each)	/1
Running the code, using a test input Does the code do what is asked in this assignment briefing when tested? Read line, print to file, read from file and print numbers to screen, calculate and print running sum, calculate and print number of occurrences of number characters. Does it check for file? (0.5 mark) Does it use the function `float mean(float sum, float N)` to find the mean, which gets printed from main()? Are the calculations correct?	/3 /1
Output to screen Is the output on the screen in a clear layout?	/0.5
Explanation Can you explain your code well?	/0.5
Total marks possible	/6

Penalties for late submission

A late submission will be any assignment not submitted to Blackboard (MOLE) by the deadline or not brought for marking in the appointment time slot.
Late submissions will incur the usual penalties of a 5% reduction in the mark for every day (or part thereof) that the assignment is late and a mark of zero for submission more than five days late. For more information see http://www.shef.ac.uk/ssid/exams/policies.

Unfair means

You are permitted to view the resources on ACS130 Blackboard (MOLE), C text books and C programs published on the internet, for insight into C programming, but you are not permitted to copy any code that you have not written to submit as your own work. You are not permitted to seek or accept help on this assignment, or code for this assignment, from any internet site or forum. You are not permitted to work with any other person on this assignment, and you are not permitted to submit any other person's algorithm or code as your own work. **Any suspicion of the use of unfair means will be investigated and may lead to penalties.** References must be provided to any other work that is used to inform the ideas for this assignment. See http://www.shef.ac.uk/ssid/exams/plagiarism for more information.

Extenuating Circumstances

If you have medical or serious personal circumstances which cause you to be unable to submit this assignment on time or which may have affected your performance, please inform me, and complete and submit an extenuating circumstances form along with documentary evidence of the circumstances. See http://www.sheffield.ac.uk/ssid/forms/circs.

Help

This assignment briefing and the lectures and resources on the ACS130 Blackboard (MOLE) course and the training in C practice sessions provide all the information that is required to complete this assignment. It is not expected that you should need to ask further questions. If you need clarifications on the assignment, then please post a question on the ACS130 Blackboard (MOLE) discussion board. The discussion board allows everyone the chance to view an answer to a question, as well as the opportunity to contribute to the discussion.