

CS 271 and 462

PA 4 - Programming Assignment 4

See the link "Creating a Makefile" in the Resources Module.

Reminder: The gcc flag for compile only (and do not create an executable) is -c. Don't use -c and -o together in the same gcc command.

Overview of This Assignment

4 Files:

- 2 C source files
- 1 header file
- 1 makefile

You must upload the files to a CS Linux host, make, then run the executable on Linux.

Once you have finished and tested the programs, you will upload them to the PA 4 assignment in Canvas.

C Programming Skills That You Are Practicing in This Assignment

1. Creating and using arrays.
2. Increasing your skills with writing functions in C.
3. Improving your understanding of makefiles and the make process.
4. Increasing your debugging skills to include syntax errors in functions and function calls.

Reminder: If you submit a program that contains a syntax error, you will receive a grade of zero for that program.

You will find the following textbook chapters helpful.

Chapter 5 - Functions

Chapter 6 – Arrays

Chapter 8 – C characters and strings

Documentation and Style (does not apply to the makefile)

- Each program must have header comments.
- Each program must have explanatory inline comments.
- Each program must follow the course guidelines for documentation and style.

Library <ctype.h> is required. https://www.tutorialspoint.com/c_standard_library/ctype_h.htm

<stdlib.h> and <time.h> are also required.

Here is the pseudocode for the main function:

Problem 1:

- Seed the random number generator with `time(NULL)`.
- Print the message "Problem 1".
- Create an array of 20 integers.
- Call the function `fillInteger` to fill the array with random numbers between -20 and 20.
- Print a meaningful heading that describes the output.
- Print the array, 10 numbers per line.
- Call function `findConsecutive`.

Problem 2:

- Print a blank line, then the message "Problem 2".
- Create an array of 50 characters.
- Call the function `fillCharacter` to fill the array with random, lowercase letters.
- Print a meaningful heading that describes the output.
- Print the array, all 50 elements on one line, with 1 space between elements.
- Call the function `findTriples`.

Problem 3:

- Print a blank line, then the message "Problem 3".
- Input a letter from the user.
Remember that you need to use the `getchar` function to input one character (not `scanf`).

https://www.tutorialspoint.com/c_standard_library/c_function_getchar.htm

You'll need to use a loop to ensure that the user enters a letter and not some other type of character. Convert the letter to lowercase.

- Call the function `countCharacter` to find out how many times the user's input character appears in the character array (the array you created in problem 2). Print the count along with meaningful text.

Problem 4:

- Print the message "Problem 4".
- Create an array of 10 float values.
- Call function `fillFloat` to initialize the elements of the array to random float values between 1.0 and 50.0
- Print a meaningful heading that describes the output.
- Print the array, all elements on one line, with two spaces between elements.
- Call function `floatMean` to find the mean of the values. Print the mean.
- Call function `floatMin` to find the minimum value in the array. Print the minimum.
- Call function `floatMax` to find the maximum value in the array. Print the maximum.

Source file: arrayfunctions.c

void fillInteger(int a[], int length, int min, int max)

- Fill the array with random integers between min and max (inclusive). You may assume that min is less than max.

void fillCharacter(char c[], int length, char start, char end)

- Fill the array with random characters between start and end (inclusive). You may assume that start is alphabetically before end.

void findConsecutive(int array[], int length)

- Examine the array. If any two elements are consecutive integers, print a message giving the array subscripts of the two elements. (Remember, subscripts start with 0.)

For example, here's an array with 10 elements. Assume that this is the array passed to the function and the length passed is 10.

```
9    11    12    10    18    20    -10    12    -3    -2
```

For this array, there are two places where two consecutive integers occur, so the function should print

```
Elements [1] and [2] are consecutive.  
Elements [8] and [9] are consecutive.
```

void findTriples (char c[], int length)

- Examine the array. If any 3 sequential elements form an alphabetic sequence, print a message with the 3 characters.

For example, assume that an array containin the following 10 characters is passed to the function along with the length of 10.

```
a b c i k j l m n x
```

There are two places where there are 3 characters in alphabetic sequence, so the function should print

```
abc  
lmn
```

int countCharacter(char c[], int length, char searchChar)

- The function should count the number of times that searchChar appears in the array and return the count.

void fillFloat(float a[], int length, float min, float max)

- Fill the array with random float values between min and max (inclusive). You may assume that min is less than max.

Hint: Each float value needs to have only one digit to the right of the decimal. Generate an integer in the between $\text{min} * 10$ and $\text{max} * 10$, then divide the integer by 10.0.

float floatMean(float array[], int length)

- Calculate and return the arithmetic mean of all of the elements in the array

float floatMin(float array[], int length)

- Calculate and return the minimum value in the array

float floatMax(float array[], int length)

- Calculate and return the maximum value in the array

Source file: `arrayfunctions.h`

Write a preprocessor wrapper using the constant name `ARRAYFUNCTIONS_H` and place the prototypes for the 9 functions inside the wrapper.

makefile

- Include an "all" target that builds an executable named pa4 (no extension).
- Write separate targets for pa4, pa4.o, and arrayfunctions.o.
- Include a "clean" target that removes all object files (all files ending with .o).

Submit 4 files: `pa4.c`, `arrayfunctions.h`, `arrayfunctions.c`, and `makefile`

(Do not zip or tar the files. Submit all 4 files with every submission.)