



Report after written exam of **PROGRAMMING TECHNIQUES**

Date: 04-07-2022

Tommaso Cerruti - s282754

Coding exercises structure

1. EXERCISE 4

I started with a while loop going from $i=0$ and terminating when $i=l$, where l is the length of the text. For each letter, I set the count to 1. If the next letter is equal to the previous, I enter in the loop that ends whenever the new character is different (or if we reached the extreme) and increment count as the number of successive occurrences of that character. After the loop, based on the value of count I choose my next action: printing the character only, if $\text{count}=1$, printing the character and its occurrences else (if the count is greater, since we set it at the beginning as equal to 1). The external while loop, therefore terminates when all the characters have been scanned.

Note: because of the inner loop, I do not have to worry if the last part of the text was hypothetically successive and therefore having forgotten its printing part.

Any further modification to the code: none, I just missed the second argument of the last print.

2. EXERCISE 5

First of all, I created a struct "bookings" having a string, the name of the day $\text{day}[N]$, and two integers: the number of bookings nbook , and the total number of people involved in the bookings count. I begin by initializing an array of type bookings, with the appropriate name (given from an array with the seven days of the week) and setting nbook and count both to zero so that they can be later incremented, hypothetically. Then, I read from the file and increment the appropriate booking in the array. Once I finish reading the file, for each day, I calculate the average number of customers each day into the variable avg . I print the text in the required form, and pass to the next day. During this printing loop, I discover what is maximum number of reservations that a day has had, the value which then has to be returned (max).

Any further modification to the code: none.

3. EXERCISE 6

First of all, I created a struct "customer" having a string, the name of the customer idv[N], and one float that will work as the total bill of the customer. I begin by setting num (the number of customers mentioned, value to be returned) to zero (which does not have to be defined since it is one of the arguments) and by initializing an array of type customer, setting to each index val equal to zero so that it will be later modified. Then I start reading the file, and I add the bill of customer C00N of that line in the array of customers at index N (I got the approval of the professor that the assumption of nomenclature C00N, where N is a integer from 0 to 9 was fair, since the total customers are 10) and I copy the name, so that for some of the 10 possible customers that were not mentioned in the file, their value of the name will stay NULL, and I can easily know which of the 10 customers was mentioned at least once. While I do this until the end of the file, I check if the bill of customer i (the one in the loop) is higher than the max (which I assumed to be equal to zero). After I finish reading the file, for each of the ten customers I check if the reference value of the id is NULL, which would mean that I never found such customer in the text. If it is not NULL, it means that I found it at least once, and therefore I print their id and val, and for each print I increment num. After, I also print the customer with the highest dept, and I end with returning num, the variable which refers to the number of customers found.

Any further modification to the code: two modifications (see the comment referencing to the underlined text above).

1. I assumed that the N in C00N is referring to the index in the customers array, however in order to do it I should have used the function atoi (from stdlib.h), since the fourth character is still a character, and I should have put it as integer in order for it to work fine as index.
2. In order to see if the string field was null, instead of using the equal operator, I should have used the strcmp function.