Structures and Strings

Exercise 1:

Consider a classroom which consists of 20 students. Each student has a name, age, and grade.

1) Create a structure of Student containing a: name, age, and grade (where grades are integers.)
2) In the main function:
   a) Read an array of Student structures, for each read a name, age, and grade.
   b) Store the students with the highest and lowest grades.
   c) Add the word `_adult` to the students' names which are ≥ 18, otherwise, add `_minor`.

Exercise 2:

Read a string, reverse it, and then sort it using an algorithm you know.

Exercise 3:

Create the `atoi()` function.
For simplicity, we will assume the number is positive. Read a number string, such as `"74621"`, and turn it into an integer under these conditions:

1) A string with no spaces or leading letters, just numbers. Ex: `"37628"` $\rightarrow$ `37628`
2) A string with leading spaces, and could contain a second number. Ex: `"37 2727"` $\rightarrow$ `37`
3) If the string contains letters, then transform only the integer part. Ex: `"367hf72"` $\rightarrow$ `367`
4) When mixing all these conditions, you get: `"74 gh63"` $\rightarrow$ `74`

Exercise 4:

1) Read a string and capitalize every first letter of each word in the string.
2) Now, count how many words exist in the string.

Exercise 5:

Imagine a library with books. Each book has a title, author, and year of submission.

1) Create a structure for each book which contains: a title string, an author string, and the year integer. In other words, an array of book structures.

2) Create a program interface that lets you choose between "Search by Title," "Search by Author," and "Search by Year." For each choice, you can choose whether to write an external function or write inside the loop.

3) For choice 1, write a system that lets you search for a book with a string inserted by the user. Note: if a book is named "The Last Flight," and the user searches for "The Last," the book will show up.

4) Choice 2 follows the same premise as choice 1. Use the same logic to search for the book by the author's name.

5) For choice 3, search for books that were written during the inserted year.

Exercise 6:

Read a string, and display the letter, or character, with the biggest frequency, alongside said frequency.

Exercise 7:

Given a string s containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string is valid.
An input string is valid if:

1) Open brackets must be closed by the same type of brackets.

2) Open brackets must be closed in the correct order.

3) Every close bracket has a corresponding open bracket of the same type.

Ex: `"{ ( }"` → Not Valid. `"{ () }"` → Valid. `""` → Not Valid. `")"` → Not Valid.