CSE102 – Computer Programming
Homework #7

Due Date: 18/04/2020

Hand in: A student with number 20180000001 should hand in a file named 20180000001.c for this homework.

Write a complete program describing all the tasks below.

Part 0: The program reads a text file named 'Video_Games.txt'. Every row of this text file includes several informations about a single game. You are asked to read every row as a string and decompose these informations which are separated by commas.

Here is the excel format of the txt file, so that you can get used to the data.

Name	Genre	Platform	Year_of_Release	NA_Sales	EU_Sales	Global_Sales	User_Score
Alter Ego	Simulation	PC	1985	0	0.03	0.03	5.8
SimCity	Simulation	PC	1988	0	0.02	0.03	2.2
Doom	Shooter	PC	1992	0.02	0	0.03	8.3
Star Wars: Dark Forces	Shooter	PC	1994	1.09	0.77	1.95	7.7
Battle Arena Toshinden	Fighting	PS	1994	0.39	0.26	1.27	6.3
Resident Evil	Action	PS	1996	2.05	1.16	5.05	9
Tekken 2	Fighting	PS	1996	2.26	1.89	5.74	8.9
Diablo	Role-Playing	PC	1996	0.01	1.58	1.59	8.7
Tomb Raider	Action	PS	1996	2.29	1.97	4.63	8.6
Tomb Raider	Action	PC	1996	0.96	0.07	1.03	8.5

- The program should keep the names of the games in a one-dimensional string array.
- The program should keep the names of the genres & platforms in two different one-dimensional strring arrays. Please be sure that the elements are not overlapping! I.e. there shouldn't be more than one 'simulation' in the array just because there are more than one simulation games.
- The program should keep the data of every single game in a two-dimensional float array. This array should include 7 different data for every game, including genre&platform. The genre/platform cell should point the index of the genre/platform array which stores the names of the related genre/platform.

    For example;

    If the genre of the game is 'simulation' and the index of the string 'simulation' in the *array of genres* is 5, then the cell which shows the genre of the game should be '5.0' in the *float array*.

    Be careful! Some of the games has no value for 'global_sales' data. They have a string 'not_available' instead of a float number. Store them in the float array wisely cause you will need that information later on!

- The float array shouldn't include any information about the name of the games since they are stored in a string array with the same indexes.

Part 1: (10 pts) The program should support 8 different operations. Therefore, there should be a menu as below. The program should terminate if and only if the user enters '8' as the input. The menu should appear *again and again* **if an input is invalid** or **if the previously requested operation is done**. If the menu appears because of an invalid input, there should be an error message too.

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation:
```

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 10

TRY AGAIN.


0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation:
```

Part 2: (20 pts) 0th and the 1st operations are almost the same. If the user enter '0' (or '1') as input, then the program should list the genres (or the platforms) as strings.

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 0


simulation
shooter
fighting
action
role_playing
racing
misc
puzzle
platform
strategy
sports
adventure
```

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 1


pc
ps
dc
ps2
x
gba
gc
ds
psp
x360
```

Part 3: (15 pts) 2nd operation lists the games according to an information received from the user. Firstly, the program asks for a 'year', after that the program asks 'until that year or since that year?'. Finally the program lists the names of the all games which released until/since that year. Don't forget that the program should handle the invalid year or period values. I.e. if the oldest game was released in 1995 and the user wants to print the game which released until 1996, the program should show an error message and asks for new values since there is no game before 1996.

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 2


Enter a year:
1995
Until (0) or Since (1)  1995: 0

alter_ego
simcity
doom
star_wars:dark_forces
battle_arena_toshinden
```

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 2


Enter a year:
2000
Until (0) or Since (1)  2000: 1
TRY AGAIN.

Enter a year:
```

Part 4: (15 Pts) The 3rd operation asks for a game name from the user. If the input matches with a name from the name array, then the program prints all information of that game. If the input doesn't match with any element of the name array, then it should give an error message and ask for a new input. If the 'global_sales' information of that game was 'not_available' when the program read it, then the users should see it as 'Not Available'.

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 3

Please enter the name of the game:
doom
Name:          doom
Genre:         shooter
Platform:      pc
Year:          1992
Sales In NA:   0.020000
Sales In EU:   0.000000
Total Sales:   0.030000
User Score:    8.300000
```

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 3

Please enter the name of the game:
tekken_3
Name:          tekken_3
Genre:         fighting
Platform:      ps
Year:          1998
Sales In NA:   3.270000
Sales In EU:   2.220000
Total Sales:   Not Available
User Score:    9.100000
```

Part 5: (10 pts) 4th operation basically calculates and prints the average of the user scores of all games.

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 4

Average: 7.805035
```

Part 6: (10 pts) 5th operation asks a game name and prints either the game was more populer in NA or EU according to the sales. If the sales are equal, specify that. Again, if the game name is not in the database, the program should ask for another name.

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 5

Please enter the name of the game:
catwoman
This game was more popular in North America.
```

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 5

Please enter the name of the game:
simcity
This game was more popular in Europe.
```

Part 7: (20 pts) 6th and 7th operations are printing the frequences of every genre/platform. I.e. they prints the number of games for every genre/platform.

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 6

simulation      93
shooter         257
fighting        125
action          359
role_playing    169
racing          225
misc            82
puzzle          24
platform        167
strategy        93
sports          336
adventure       70
```

```
0: List of the Genres
1: List of the Platforms
2: List of the Games Through the Years
3: All Informations of a Single Game
4: Average of the User Scores
5: Georographical Information of a Single Game
6: Frequence of the Genres
7: Frequence of the Platforms
8: Exit

Please select an operation: 7

pc      121
ps      114
dc      8
ps2     775
x       432
gba     213
gc      298
ds      19
psp     18
x360    2
```

❖ You can assume that;
  ➢ The longest game name has 100 characters
  ➢ The longest genre or platform name has 20 characters.
  ➢ The longest row of the text file has 200 characters

❖ You are not allowed to use any library other than;
  ➢ <stdio.h>
  ➢ <string.h>
  ➢ <stdlib.h>
❖ You can use the 'strtod' function from <stdlib.h> library to parse a string to float.
❖ Some games are released on different platforms, which means some of the game names are the same. When you are search for a game with the name of it, pick the first one you find.
❖ You can write your own functions to make things easier.
❖ Don't forget that the program shouldn't terminate when an operation is done.

General Rules:

1. The program must be developed on Linux based OS and must be compiled with GCC compiler, any problem which rises due to using another OS or compiler won't be tolerated.
2. Note that if any part of your program is not working as expected, then you can get zero from the related part, even it's working in some way.
3. Upload your .c file on to Moodle to deliver your homework. Name format can be found on the top of this homework sheet.
4. You can ask any question about the homework by sending an email to sgulmez2018@gtu.edu.tr or by using the forum in the Moodle page of the course.