STAGE 1

FINAL PROJECT

The final project consists of three parts:

1st: Hacker rank.

2nd: Company' database.

3rd: Paint program.

FIRST PART:

Objectives:

- Practice C coding more.
- Get familiar with a well-known coding platform.
- Get used to using C for solving various problems.
- Practice problem solving.

Problem Statement:

It is required to get 200 points "3 stars" or more on **Hacker rank** for problem solving and 200 points "4 stars" for C Programming.

Each challenger has to create an account on Hacker rank website then start programming in **C** and practice problem solving problems**.**

Link: https://www.hackerrank.com/

Submission:

- A screen shot of your newly created account with score of ZERO and a date starting from today.
- The screenshot should be sent to: microcoders.alex@gmail.com
- The subject of the e-mail should be "Final Project: HackerRank Account".
- You should mention your ID in the mail.
- For those who didn't join the first stage, mention your exam ID. You will get your new ID in a reply.
- The deadline is Tuesday 30/7 at 23:59.
- On the discussion day, you will be asked to open your account to be viewed and check that 475 are achieved.

Grading:

This part acts only as a condition for accepting the whole project. It doesn't affect the total grade.

SECOND PART:

Objectives:

- Upgrade your skills of coding with functions, structures.
- Implement searching and sorting algorithms.
- Practice creating user interface programs.

Problem Statement:

It is required to write a program that will create and maintain the data base of a company.

The data base includes five fields which are the names of the employees, their phone numbers, their e-mails, their birth dates and their current job titles.

The program should enable the user to use the following functions:

1. **ADD**: Using this function, the user can add the data of a new employee field by field.
2. **LOAD**: Using this function, the user can add the data of a new employee by reading it from a text file. The data should be written as: "Adam Mohamed, 01122334455, AdamMohamed.19@gmail.com, 1/1/1980, Sales Manager" Where each field is separated by a comma from the one before.
3. **DELETE**: Using this function, the user can delete the data of any employee by just specifying his name.
4. **SAVE**: Using this function, the user can write the whole data base in a text file with the same format mentioned in the LOAD function. The data should be printed alphabetically ordered, from A to Z.

5. **SEARCH**: Using this function, the user can get the data of any employee by just knowing his name. Data should be printed on the console.
6. **MODIFY**: Using this function, the user can modify the data of any employee by being asked about his name and required field to be modified. The name and birthdate shouldn't be allowed to be changed.
7. **QUIT**: Using this function, the user can exit the program. However, he should be asked first wither to save, exit without saving or staying.

Bonus

8. **SORT**: Using this function, the user can arrange the database text file either by arranging the data according to names from A to Z or from Z to A or according to age, birthdate, from youngest to eldest or from eldest to youngest.

Note that:

1. The program should validate the data entered for each field and print a specified error if any one isn't valid:
   - Validate that the number of fields is the same as required.
   - Validate the Name as string of letters only.
   - Validate phone numbers as numbers only.
   - Validate the mail: Something@website.com
     Where "website" is either gmail, yahoo or other website. Also, "Something" doesn't start with a number.
   - Validate the following format for date DD/MM/YYYY, where YYYY should be a number from 1959 to 1996, MM a number from 01 to 12 and DD a number from 1

to 31 for months 1, 3, 5, 7, 8, 10 and 12, a number from 1 to 29 for month 2 and from 1 to 30 for the rest. Bonus: check for leap years.
- Validate Job title as one from the titles in a text file.

2. The program should print meaningful messages in case of errors. Ex: "Name doesn't exist" in case of searching for a name not found in the data base.
3. A brief manual on how to use the program should be available for the user at any time.
4. You are free to create whatever company you want and job titles. A text file of all the information on the company you created should be created.
5. The data of each employee should be between " ". Any spaces between any data except names and job titles should be ignored. Anything between the first user" and the second user "should be ignored. Ex: "Adam Mohamed, 01122334455, AdamMohamed.19@gmail.com, 1/1/1980, Sales Manager" bla bla"Amir Ali, 09988776655, AmirAli.5@yahoo.com, 12/12/1990, Software developer"

Submission:

- Only one zipped folder should be sent.
- The folder should be named Project1.2_ID, where ID is your new ID. Ex: Project1.2_01.
- The folder should contain:
    i. A C File: CompanyDatabase.c
    ii. A text File: CompanyDatabase.txt
    The file should contain at least 10 employees.
    iii. A text File: CompanyInfo.txt
    The file should contain the name of the company and the job titles.

- The project should be sent to: microcoders.alex@gmail.com
- The subject of the e-mail should be “Project1: Company’s data base”.
- The deadline is Thursday 8/8 at 23:59.
- On the discussion day, you will be asked to run your code and explain any part of it.

Grading:

This part has a total grade of fifty, half the total grade of the project.

Each bonus is rewarded 5 points.

THIRD PART:

Objectives:

- Learn how to create a program where you have a full command of the console.
- Learn how to use the mouse in the console.
- Improve your skills in creating user interface programs.
- Get introduced to different standard libraries.
- Use C to create any desired shapes.

Problem Statement:

It is required to create a Paint on console program. When the user opens the program, a menu appears where he is asked to choose Free paint, drawing geometrical figures or exiting the program.

1. **Free paint**:
   The user should be able to:
   a. Draw whatever he wants using the mouse.
   b. Erase any part of what he drew using mouse.
   c. Choose any color to draw with "at least 5 colors".
   d. Change the color of what he had drawn.
   e. Clear the whole screen by pressing a Clear button using the mouse.
   f. Return to the main menu by pressing a Home button using the mouse.

2. **Geometrical Figures**:
   The user should be able to:
   a. Draw geometrical figures as lines, squares, rectangles, triangles and circles using keyboard.
   b. Specify the dimensions of each figure using keyboard.
   c. Choose any color to draw with using keyboard.

d. Move the figure he had drawn.
e. Change the color of the figure he had drawn using keyboard.
f. Clear the whole screen by pressing a Clear button using the mouse.
g. Return to the main menu by pressing a Home button using the mouse.

3. **Exit**:
   If pressed, the user is asked "Are you sure you want to quit?" and he choose to press either Yes or No buttons using mouse.

Submission:

- Only one zipped folder should be sent.
- The folder should be named Project1.3_ TeamID, where TeamID is your Team ID. Ex: Project1.3_01.
- The folder should contain:
  i. A C File: PaintOnConsole.c
  ii. A text File: Manual.txt
  The file should explain how to use the program.
- The project should be sent to: microcoders.alex@gmail.com
- The subject of the e-mail should be "Project1: Paint on console".
- The deadline is Thursday 8/8 at 23:59.
- On the discussion day: You will be asked to run your code and explain any part of it.

Grading:

This part has a total grade of fifty, half the total grade of the project.

The grade will be on how easily the user can interface with the program and how much can he do with the program.

Creative additions for the program will grant you a bonus.

Important Notes:

- The discussion date will be announced later.
- First and second part of the project is done individually.
- The third part of the project, Print on console, will be done in groups of 5.
- Each member of the team will be asked and graded individually to ensure the participation of each member in the project.