# NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES, KARACHI,

# School of Computing

**ASSIGNMENT NO 03**

|  |
| --- |
| **Subject: Programming Fundamentals -CS1002-FALL2022 Post Date: 20/11/2022**  **Total Marks: 40 Due Date: 27/11/2022, 04:00 pm**  **Course Instructor: Shahbaz Siddiqui** |

**Instructions to be strictly followed.**

* Each student should submit these files:
  + ***A zip of all .C files named as "A3-Q#[StudentID].c" where # is the question number and Student ID is your ID.***
  + ***A DOC file where they copy code for each code and screen shot of the output. This document contains all the questions, answer codes and output in sequence. Name this document as “A3-[StudentID].docx”.***
* Each output should have STUDENT ID and NAME of the student at the top.
* **Viva will be conducted randomly from the assignment. Hence, make sure you have developed logic and have clear idea about each submitted solution**.
* Zero grade for plagiarism (copy/ cheating) and late submissions.

**Question 1. [10 Marks]**

One of the master coders designed a subscript block that stores random characters in a multidimensional array. You are asked to design a program that will find a given string in a multidimensional array of characters. The search for characters can be present and operational from left to right and top to down only. The program should create a 6 x 5 2D array and populate it with random alphabet characters. After that, the program should print it in a tabular form as shown below.

Search the user-entered string in the 2D array, if it is present then add a point to the score, if it is not available then subtract one. Print the score at every input. The program stops asking and re-populates the 2D array with new random characters when the user enters “END” as the string.

[NOTE: After generating random numbers, the last row should have your student ID’s last four digits]

Example:

Array:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E | D | D | F | R | R |
| A | F | V | A | Q | R |
| T | E | B | S | T | A |
| L | J | G | T | T | A |
| 1 | 2 | 3 | 4 | Q | K |

Search Str= ‘FAST’

Output:

FAST is present🡪 Score 1

EAT is present🡪 Score 2

GREAT is not present 🡪 Score 1

**Question 2. [10 Marks]**

You are invited to participate in an encryption algorithm competition. Encryption is a method to change the formation of the message using pre-defined rules. After encryption, the message becomes secure and illegible. Develop a program that takes user input as a string sentence and prints its encryption version using the following rules:

* You must include your ID, name and section number as sample input. The program should also include your ID as a prefix to all input sentences.
* Each word should be encrypted separately.
* If a word starts with consonant letter, then that word should be shifted two places in circular way. For example, “trend” become “ndtre”, “brown” become “wnbro”, etc. Also, after shifting, do append “ay” at the end of the word. So, a word “trend” become “ndtreay”.
* If a word starts with a vowel letter, then it should be encrypted by simply appending "way". For example, “end” becomes “endway”.
* The letter 'y' is a treated as a consonant if it appears as the first letter of a word (yard -> ardyay). As a vowel if it appears at an interior location (at mid or somewhere) take consonants before the first vowel shift to end. Eg: crybaby-> abycrybay
* Also, "sch" at the beginning should become "sk" at the end. Eg: schooner -> oonerskay
* Capitalized words should be capitalized after transformation. Eg: Sam -> Amsay

A sample input is:

“Brown fox jumps a yard.” after encryption “22k-1234 Wnbroay oxfay psjumay away rdyaay.”

**Question 3. [20 Marks]**

Develop a system for the class teacher to store and retrieve data for PF course. The data includes student id, name, mid1, mid2, final, assignment and quiz score. You must use efficient data structure to store and retrieve this information. Let’s consider that your ID’s last three digits define the number of students in the class. Do print your actual student ID, name and function name whenever a user defined function is called.

[Note: For example, your id is 21k-1234, which means there are 234 students in the class. To test your code, you can consider that total students in your class are 10. After developing all the solutions, you can change the number of students to last three digits of your ID.]

1. Initialize appropriate variables, with proper messages for the user and take input for each student’s information and store for later use. Also, you are required to check student ID of each new input to check that user already exists or not. If user already exists, then print “Student ID exists, please enter data for another student.”
2. After all inputs, show a menu to the user as given below:

Enter appropriate letter for your choice of action:

|  |  |
| --- | --- |
| g | for printing GPA of a given student ID. This option will ask user for a student ID and will perform linear search for the said ID. After that, it should use accumulateEvaluations*() and getGPA()* to print the student ID, name and GPA of the student. |
| a | for printing GPA of all students in a tabular form. This option will use accumulateEvaluations*() and getGPA()* in such a way that it prints all the data of all students in a tabular form. A sample output is given below:  ID Name GPA  21k-1234 Ayan 3.0  21k-2234 Ali 1.5  … |
| r | for replacing the details of the student at the 7th index by your exact data. Your ID, name and evaluations should be correct and up to date. The remaining evaluation can be entered as you anticipate it. |
| f | for finding the details of a student using *smartFind()* function |
| s | for printing the statistics of the class. It should print the following:  Total student count is …  Average score mid1…  Average score mid 2…  Average score final…  Average total score  Student ID … has the maximum score of …  List of student ID is given below who didn’t pass the course  ...  …  “Thank you, my ID is 22k-1234 and my name is Ayaan” |
| e | Exit the program |

1. Write a function accumulateEvaluations*()* that takes input mid1, mid2, final, assignment and quiz score for single student and returns the GPA. For GPA take help from function in part C.
2. Write a function *getGPA()* that takes student score and total score and return the GPA. Percentage to GPA calculation should reflect the FAST-NU system. (Study and get exact formula to calculate GPA for a course using percentage).
3. Write a *smartFind()* function that takes a string as an input and finds student record using that string. The input string can be student ID or Student name. Your defined function should be smart enough to correctly search for appropriate record and print all the details of the student. For example, if a user enter 21k-1234 or Aayan, both terms should be able to find the same record and print all the details.