CS19003 Programming and Data Structures Laboratory
Assignment 6

- ▷ Name the files as ⟨assignment no⟩_⟨question no⟩_⟨roll no⟩.c, without the ⟨ and ⟩. Consult your mentor for any confusion. Penalty if the file names do not stick to this convention.
- ▷ This is an assignment on two-dimensional arrays, structures and pointers. Do not use dynamic memory allocation or any other advanced concept.

Please take note of the following:

- ▷ The structure declarations (for question 1) and names of variables and functions should be exactly as given in the questions. Any deviation will attract penalty.
- ▷ The format of the output should match as closely with the sample output as possible. Significant deviation will attract penalty. In the event of any confusion, ask us.
- ▷ Global variables (including structure variables in question 1) are NOT allowed. However, the structure definitions in question 1 will be global.
- ▷ Use of string library functions is NOT allowed.
- ▷ Read the questions carefully. Any deviation from the specifications mentioned will attract penalty. In the event of any confusion, ask us.

---

1. In this program you will develop an interactive program to maintain a database for tutors.

   - ▷ Define the following two global structures. The details of the fields are written as comments in the definiton.

     struct Tutor{

     *char name[20]; /** only English letters, no space */

     *char ID[20]; /** alphanumeric (i.e. letters and digits), no space */

     *char subject[20][20]; /**each row contains a subject as a null-terminated string of letters (no space). An empty string denotes no subject. If a tutor can teach 8 subjects, then the first 8 rows should hold the names of those subjects. The remaining rows should contain empty strings. */

     *int score; /** An integer between 0 and 100, inclusive, computed as an average of student feedbacks. */

     };

     struct *TutorDB*{

     *struct Tutor T[100]; /** An array of tutor records */

     *int n; /** Number of records in the database */

     };

- Make the following structure variable declaration in *main* (*Note: do not define a global structure variable*).

  struct TutorDB L;

- Write a function named *view* with appropriate prototype that displays the details of all the tutor records in the variable *L* in the order in which they were inserted (or reports if there is no record). Subjects should be displayed as a space-separated list.
- Write a function named *insert* with appropriate prototype that asks the user for details of a new tutor, and inserts a new tutor record in L. You may assume that the number of existing records is strictly less than 200.
- In *main*, ask the user to press 1 to view the database, 2 to insert a new record in the database and 3 to exit. If the user presses 1, display the database using the function *view* and return to the above menu. If the user presses 2, call the fuction *insert* to ask the user for the details of the new tutor and create a new record in the database, and then return to the above menu. If the user presses 3, the program terminates. If the user presses anything else, display "Invalid input" and return to the menu.

Sample input/output:

```
Press:
1: View database
2: Insert tutor
3: exit
1
Database empty
Press:
1: View database
2: Insert tutor
3: exit
2
Enter name: Sudeep
Enter ID: 47UYT
How many subjects? 2
Enter subject 1: Physics
Enter subject 2: Economics
Enter score: 89
Press:
1: View database
2: Insert tutor
3: exit
1
Name: Sudeep
```

```
ID: 47UYT
Subjects: Physics Economics
Score: 89
Press:
1: View database
2: Insert tutor
3: exit
2
Enter name: Taufiq
Enter ID: 67H78
How many subjects? 3
Enter subject 1: Programming
Enter subject 2: Statistics
Enter subject 3: Business
Enter score: 87
Press:
1: View database
2: Insert tutor
3: exit
1
Name: Sudeep
ID: 47UYT
Subjects: Physics Economics
Score: 89
Name: Taufiq
ID: 67H78
Subjects: Programming Statistics Business
Score: 87
Press:
1: View database
2: Insert tutor
3: exit
5
Invalid input
Press:
1: View database
```

```
2: Insert tutor
3: exit
3
```

[100 points]

2. In the main

   - Declare a 2-dimensional array A with 20 rows and 10 columns.
   - Ask the user for some positive (thus, non-zero) integers. Assume that the number of numbers entered is at most 20. Aso assume that each integer has at most 9 digits.
   - Scan each integer into a variable of an integer type large enough to hold a 9 digit number. In my computer, the *int* type will suffice.
   - Store the i-th number that the user enters in the i-th row of the array A as follows. The first cell of the i-th row should contain the number of digits of the number. From the second cell onwards, the digits of the numbers will be stored. For example, the number 55705 will be stored in a row as follows.

     [5 5 5 7 0 5 x x x x]

     The cells marked above by 'x' can contain any number. If the number of numbers that the user enters is less than 20, then the latter rows, which do not hold any number, could contain arbitrary entries (eg. if the user enters two numbers, only first two rows will be relevant). Note that to store an integer in a row in this manner you will have to extract the digits from the integer.
   - After creating the matrix A, print it. In the sample output below, the matrix has a lot of random entries. Your random entries may be different. However, make sure that the entries that matter match.

   Write a function named *Add* with appropriate prototype so that it can be called from the main as *Add(A, n)*. Here, A is the array with the input numbers stored in the rows as described above, and n is the number of numbers that the user enters. The function should print the sum of the numbers that the user enters. Assume that the sum is also at most 9 digits long.

   Sample input/output:

```
How many numbers? 4
Enter number 1: 243
Enter number 2: 12
Enter number 3: 45112
Enter number 4: 3421
The matrix
3 2 4 3 4196154 0 4196864 0 1548331760 32655
2 1 2 0 1548265304 32655 -1 0 8064 65535
5 4 5 1 1 2 3948576 0 1009492208 32767
4 3 4 2 1 0 0 0 1548303940 32655
Sum is 48788
```

[100 points]