National Institute of Technology Calicut
Department of Computer Science and Engineering
B. Tech. (CSE) – First Semester
CS1091E: Programming Laboratory
Problem Set – 7

Submission deadline (on or before):

- 07/11/23, 5:00 PM

Policies for Submission and Evaluation:

- You must submit your programs in the moodle (Eduserver) course page, on or before the submission deadline. Also, ensure that your programs compile and execute without errors in the linux platform. During evaluation, failure to execute programs without compilation errors may lead to zero marks for that program. Detection of ANY malpractice can lead to awarding an F grade in the course.

Naming Conventions for Individual Program

- $PS < PROBLEM\_SET\_NUMBER > \_ < ROLLNO > \_ < FIRST - NAME > \_ < PROGRAM - NUMBER > . < extension >$ (For example: $PS07\_BxxyyyyCS\_LAXMAN\_1.c$). Please make sure that you follow the naming conventions correctly.

Naming Conventions for Submission

- Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar, .tar, .gz) containing the source code (.c file). The name of this file must be $PS < PROBLEM\_SET\_NUMBER > \_ < ROLLNO > \_ < FIRST - NAME > .zip$ (For example: $PS07\_BxxyyyyCS\_LAXMAN.zip$). DO NOT add any other files (like temporary files, input files, etc.) except your source code, into the zip archive.

Standard of Conduct

- Violations of academic integrity will be severely penalized. Each student is expected to adhere to high standards of ethical conduct, especially those related to cheating and plagiarism. Any submitted work MUST BE an individual effort. Any academic dishonesty will result in zero marks in the corresponding exam or evaluation and will be reported to the department council for record keeping and for permission to assign F grade in the course.

General Instructions

- Programs should be written in C language and compiled using C compiler in Linux platform. Sample inputs are just indicative. **Submit the**

solutions to questions 1 and 2 as a single .zip file through the submission link in Eduserver.

QUESTIONS

1. Read the marks (**int** type) of n ($5 <= n <= 100$ is an integer) students and assign a character grade to each student based on the grading criteria given below:

Marks	Grade
80 -100	A
60 -79	B
50 -59	P
0 -49	F

The value of n is entered by the user. Use two one dimensional arrays, say M and G, M for storing the marks and G for storing the grades. Define the functions whose prototypes are given below:

char marksToGrade(int); //given marks, returns the corresponding grade

void printGradeCount(void); //prints for each grade, the number of
//students who scored that particular grade

int gradeCount (char); //given a grade, counts the number of that
//particular grade in G and returns the count

The *main()* function, after creating the array M (by reading in the marks), should create array G by calling *marksToGrade()* for each element in M. Then *main()* should invoke *printGradeCount()* which in turn should invoke *gradeCount()* to get the count of each grade in G.

Input format:
<n>
$< mark_0 > < mark_1 > ... < mark_{n-1} >$ //$mark_i$ is the mark of the i^{th} //student

Output format:

Grade	Count
A	<number of A grades>
B	<number of B grades>
P	<number of P grades>
F	<number of F grades>

2. Read the marks (**int** type) of n ($5 <= n <= 100$ is an integer) students in three different courses and compute the total mark for each student (given the corresponding row index), and highest, lowest, average marks (**float** type) for each course. Use 2 Dimensional arrays for storing the marks. Use separate functions for computing the highest mark, lowest mark and average mark for a given course. Each function should take the course id (0 / 1 / 2) as argument (course id is the column number of the corresponding column in the 2 Dimensional Marks array). Write a menu driven program. After creating the array, program should display a menu as follows:

Please enter a number indicating your choice
1: highest mark in a course
2: lowest mark in a course
3: average marks for a course
4: total marks of a student
0: exit
As per the user's choice, the program should compute and print the required value. It should be possible to enter choices any number of times, in a single run of the program.

Input format:
<n>
$< mark_{00} > < mark_{01} > < mark_{02} >$
$< mark_{10} > < mark_{11} > < mark_{12} >$
.....
$< mark_{n-1\ 0} > < mark_{n-1\ 1} > < mark_{n-1\ 2} >$
// $mark_{ij} (0 <= i <= n-1\ \&\ 0 <= j <= 2)$ is the mark of student i in course j

Output format:
As per the sample input / output given below:
3
62 35 49
69 68 72
74 88 55

Please enter a number indicating your choice
1: highest mark in a course
2: lowest mark in a course
3: average marks for a course
4: total marks of a student
0: exit

2
Enter course ID:0
Lowest mark in course 0 : 62

Please enter a number indicating your choice
1: highest mark in a course
2: lowest mark in a course
3: average marks for a course
4: total marks of a student
0: exit

1
Enter course ID:1
Highest mark in course 1 : 88

Please enter a number indicating your choice
1: highest mark in a course
2: lowest mark in a course
3: average marks for a course
4: total marks of a student
0: exit

0

3. (Practice question, do not submit)
   Represent matrices using 2D arrays. Implement matrix operations - addition, multiplication and transpose.