

National Institute of Technology Calicut
Department of Computer Science and Engineering
B. Tech. (CSE) – First Semester
CS1091E: Programming Laboratory
Problem Set – 9

Submission deadline (on or before):

- 21/11/23, 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: *PS09_BxxxxxyCS_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: *PS09_BxxxxxyCS_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 2 and 3 as a single .zip file through the submission link in Eduserver.

QUESTIONS

1. **(Practice question - do not submit)** Implement all example programs/program fragments given in the lecture materials on pointers posted in the theory course page.
2. Read the elements in an integer array A of size 5. Find the smallest element in the array and interchange the smallest element and the element at 0th position. The program should not use global variables. Define the functions whose prototypes are given below:

```
void swap(int *a, int *b) //swaps the values in the variables pointed to
                           //by a and b.
int smallest(int a[]) //returns the position of the smallest element in a
```

The main function should first read the 5 elements of A and then invoke *smallest()* followed by *swap()*. *main()* should then print the array elements - all elements in the same line with a single space between each element, as shown in the sample o/p.

```
i/p:
30 20 10 50 40
o/p:
10 20 30 50 40
```

3. Read the marks (**int** type) of n ($1 \leq n \leq 100$ 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. The marks and grades are to be stored in two one dimensional arrays M and G , which are local to $main()$. The program should not use global variables. Define the functions whose prototypes are given below:

```
void readMarks(int a[], int n);
    //reads n marks and stores them in array a
void marksToGrade(int a[], char b[], n);
    //converts each mark in a to a letter grade and
    //stores the grade in the corresponding position in b
    //n is the number of marks in a
void printGradeCount(char a[], int n);
    //prints for each grade, in the order A, B, P, F,
    //the number of students who scored that particular grade
    //a contains the grades of n students
```

The $main()$ function, after creating the array M (by invoking $readMarks()$), should create array G by calling $marksToGrade()$. Then $main()$ should invoke $printGradeCount()$ to print the count of each grade in G .

Input format: (no prompting messages for input)

```
< n >
< mark0 > < mark1 > ... < markn-1 >
    //marki is the mark of the  $i^{th}$  student
```

Output format: (a single tab space between each grade and count)

```
A    <number of A grades>
B    <number of B grades>
P    <number of P grades>
F    <number of F grades>
```