National Institute of Technology Calicut
Department of Computer Science and Engineering
Third Semester B. Tech.(CSE)
CS2092D Programming Laboratory
Assignment #4

Submission deadline (on or before):

- 30.09.2020, 10:00 PM

Policies for Submission and Evaluation:

- You must submit your assignment in the Eduserver course page, on or before the submission deadline.
- Ensure that your programs will compile and execute without errors in the Linux platform.
- During the evaluation, failure to execute programs without compilation errors may lead to zero marks for that evaluation.
- Detection of ANY malpractice related to the lab course can lead to awarding an F grade in the course.

Naming Conventions for Submission

- Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar, .tar, .gz). The name of this file must be named as

  `ASSG<NUMBER>_<ROLLNO>_<FIRST-NAME>.zip`

  (For example: *ASSG*1_*BxxyyyyCS_LAXMAN.zip*). DO NOT add any other files (like temporary files, input files, etc.) or folders, except your source code, into the zip archive.

- The source codes must be named as

  `ASSG<NUMBER>_<ROLLNO>_<FIRST-NAME>_<PROGRAM-NUMBER>.c`

  (For example: *ASSG*1_*BxxyyyyCS_LAXMAN*_1.*c*). If you do not conform to the above naming conventions, your submission might not be recognized by our automated tools, and hence will lead to a score of 0 marks for the submission. So, make sure that you follow the naming conventions.

Standard of Conduct

- Violation 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. The department policy on academic integrity can be found at: http://cse.nitc.ac.in/sites/default/files/Academic-Integrity_new.pdf.

General Instructions

- Programs should be written in C language and compiled using C compiler in Linux platform. **Submit the solution to question 1 through the submission link in Eduserver. You should complete the other two programs before the lab session. During the evaluation we may be asking you to modify any of these three programs.**

The Sorting Problem can be formally stated in terms of the input/output relationship as follows:

Input: A sequence of n numbers $\langle a_1, a_2, \ldots, a_n \rangle$

Output: A permutation $\langle a'_1, a'_2, \ldots, a'_n \rangle$ of the input sequence such that $a'_1 \leq a'_2 \leq \cdots \leq a'_n$.

QUESTIONS

General Note: For the following program, do not declare the array as a global variable. You may pass the arrays as function arguments by using the concept of pointers.

1. Write a program that uses the QUICK-SORT algorithm for sorting a given input sequence of integers present in an array A and prints the number of comparisons performed during sorting. Your program must contain the following functions: (the notation $A[i..j]$ denotes the sub-array of A, contained within the i^{th} and j^{th} indices, both inclusive)

   - A recursive function QUICK-SORT($A,\ p,\ r$) that takes as input an array A and sorts the sub-array $A[p..r]$ using Quick-Sort.
   - A function PARTITION($A,\ p,\ q,\ r$) that takes as input an array A and partitions it into two sub arrays $A[p..q-1]$ and $A[q+1..r]$ such that each element of $A[p..q-1]$ is less than or equal to $A[q]$ which is, in turn, less than or equal to each element of $A[q+1..r]$.
   - PRINT($A,\ n$) - A function that takes as input an array A and an integer n, the size of A. It then prints the contents of A in order, with a single space separating the elements. This function should only be called from the MAIN() function.

   Input format:

   - The first line of the input contains an integer $n \in [0, 10^5]$, the size of the array A.
   - The second line lists the n elements in A, as space-separated integers in the range $[-10^3, 10^3]$.

   Output Format:

   - The first line of the output contains the elements of A in sorted order, separated by space.
   - The second line of the output contains the number of comparisons performed during sorting.

   Note:

   The number of comparisons made by Quick-Sort is highly dependent on its implementation. As such, we will be considering the number of comparisons as per the algorithm given in CLRS.

   Sample Input 1:

   7
   1 2 3 4 5 6 7

   Sample Output 1:

   1 2 3 4 5 6 7
   21

   Sample Input 2:

   7
   1 2 5 7 6 9 8

   Sample Output 2:

   1 2 5 6 7 8 9
   13

2. Write a program that reads a string of characters **A** and check whether a given substring **B** is present in **A**. Assume that the array index starts from 0. If the substring **B** is present in **A**, print the first index of **B** in **A**. If the substring **B** is present more than once in **A** then print the first index of first occurrence of the substring **B** in **A**. Otherwise, print -1.

   Input format:

   - The first line of the input contains a string **A**.
   - The second line contains a string **B**. Check whether this string **B** is a substring of string **A**.

   Output format:

   - If **B** is present in **A**, print the first index of **B** in **A**.
   - If **B** is not present in **A**, print **-1**. Assume that array index starts from **0**.

   Sample Input 1:
   Good Morning
   Morning
   Sample Output 1:
   5

   Sample Input 2:
   NIT calicut
   Kozhikode
   Sample Output 2:
   -1

3. Create an array of structures that stores the following details of each employee:
   NAME
   SALARY
   WORK PER DAY(in hours)

   Assume that the number of employees is 10 and each employee will get a Basic Salary. Increase the salary of each employee depending on the number of hours of work per day as follows :

Work per day (in hours)	8	10	>10
Increase in salary	Rs.50	Rs.100	Rs.150

   Write a menu driven program that uses the structure mentioned above and allows the user to perform the following operations:

   (a) Add an employee record
   (b) Display the details of all the employees who did not get any increment in salary
   (c) Display the details of all the employees with their final salaries.
   (d) Display the details of all the employees, given work per day(in hours).

Your program should implement the following functions:

- **main()** - repeatedly reads a character **'r', 'i', 'f' ,'w'** or **'t'** from the terminal and calls the sub-functions appropriately until character 't' is entered.
- **INSERT()**- Add an employee record.
- **PRINT_NOINCREMENT()**- Display the details of all the employees who did not get any increment in salary.
- **PRINT_FINALSALARY()**- Display the details of all the employees with their final salaries.

- **PRINT_WORKPERDAY()**- Display the details of all the employees with given work per day(in hours)

Input Format

- First line contains a character from **'r', 'i', 'f' ,'w','t'** followed by at most one integer.
- Character 'r' is to read the details of an employee. Character 'r' should be followed by a string, specifying the name of the employee. In this case, the second line contains an integer, specifying the salary of the employee. Third line contains an integer, specifying *work per day* of the employee.
- Character **'i'** is to display the details of all the employees who did not get any increment in salary.
- Character 'f' is to display the details of all the employees with their final salaries.
- Character 'w' is to display the details of all the employees, with the given *work per day*(in hours). Character 'w' should be followed by an integer, specifying the *work per day*.
- Character 't' terminates the program.

Output Format

- For option 'i', print details of all the employees who did not get any increment in salary. In this case, all the details of an employee is printed on the same line separated by a space. Order of employee details is:

  name_of_employee salary work_per_day

  If there are multiple employees, print each employee detail in a newline.

- For option 'f', print the details of all the employees with their final salaries. In this case, the name and salary of the employee is printed on the same line separated by a space. Order of employee details is:

  name_of_employee salary

  If there are multiple employees, print each employee detail in a newline.

- For option 'w', print the details of all the employees with given work per day(in hours). In this case, the name and salary of the employee is printed on the same line separated by a space. Order of employee details is:

  name_of_employee salary

  If there are multiple employees, print each employee detail in a newline.

Sample Input 1:

```
r Alice
63000
8
r Bob
63000
10
r Jack
63000
7
i
f
w 8
t
```

Sample Output:

```
Jack 63000
Alice 63050
```

Bob 63100
Jack 63000
Alice 63050