athenahealth – Programming Round' 12

Duration: 1 hour

Instructions to be followed:

- Question No. 1 is mandatory and must be attempted first.
- Save all the programs inside a folder named "<YourFullName>_<Course>_<DeptName>_<CollegeName>" on your desktop.
- Ex: Ramesh_BTech_IT_SSN
    - Programs should be named like Program1.c/Program1.cpp, Program2.c/Program2.cpp etc.

---

1. We are developing a system to rank numbers. The ranking system is based on the number of distinct divisors a number has. Given a set S of positive integers, arrange the numbers based on the number of their divisors. A number with the smallest divisor count should appear first, the one with the second smallest, comes second, and so on. If two numbers have the same number of divisors, the smaller of them comes first.

   Input: A single line with space separated positive integers. The first on the line is the size of set S. Then the elements of S follow.
   Output: The elements of S in the required order, each separated by a space.

   Sample Input:
   5 18 24 10 14 25

   Sample Output:
   25 10 14 18 24

2. A machine is designed to sort distinct numbers given as a list (say L). The instruction set of this machine has only one instruction named MOVETOLAST that takes one element of L as a parameter, it removes that element from L and appends it to the end of L. Determine the minimum number of MOVETOLAST instructions this machine would require to sort a given list.

   Input: A single of space separated integers. The first on the line is the size of list L. The elements of L follow.
   Output: A single line with the minimum number of MOVETOLAST instructions to sort L.

   Sample Input:
   4 3 1 2 4

   Sample Output:
   2

Explanation:
Initially, L is [3, 1, 2, 4]. An optimal instruction sequence is as follows.

"MOVETOLAST 3" which results in the list [1, 2, 4, 3].
"MOVETOLAST 4" which results in the sorted list [1, 2, 3, 4].

Thus, a total of 2 instructions would suffice to sort the given list.

3. Technical lead of a development team has to send out a meeting invite to all his team members to discuss on a project they are working on. There are N people in his team (including himself). He has the calendar of all the meeting participants, i.e. he knows when each member is busy. With this information, he has to schedule a meeting for T minutes, i.e. find a slot of T minutes during which all participants are free.

   Input: A single line of input. The first integer on the line represents T, the second integer represents N (N >= 2). Then N pairs of strings follows, each representing a period in time of a particular day during which that member is busy. The period will be of the format "HHMM HHMM". For example, if a pair reads, "0900 1500", then that member is busy from 9AM to 3PM during that day. Note that the time is in 24-hour format. Note that the working hours of all participants are between 9AM and 9PM.

   Output: Output the number of different slots (of T minutes) during which the meeting can be scheduled.

   Sample Input:
   30 2 0900 2029 0900 2029

   Sample Output:
   2

   Explanation: Here, T = 30 minutes. There are 2 participants and the periods during which they are busy are as follows.

   1st member -> 0900 2029 -> 9AM to 20:29PM
   2nd member -> 0900 2029 -> 9AM to 20:29PM

   The two available slots of 30 minutes are: 20:30 to 20:59 and 20:31 to 21:00.