

CS2.201: Computer Systems Organization

Spring 2025

International Institute of Information Technology, Hyderabad

CSO Lab Exam Questions

Note: Read the information given below carefully.

- There are 12 problems in this question bank.
- The first 6 questions are from the Easy category and the next 6 are from the Hard category. You will be asked to solve one question from each category in the exam.
- During the exam you should attempt only those questions which have been assigned to you.
- Assume signed/unsigned long long int or double based on the question.
- Comments: Not necessary. But some comments to guide us in evaluations would help. Note: Only some basic comments are enough. Don't add comments for each line. It would take too much of your time and would actually make it difficult for us to find the main parts of your code. So, if adding comments, just add a few comments near the main parts of your code.
- A naive solution is fine, unless the question asks you to implement a solution of a specific time complexity (for such questions, not following the time complexity requirement will result in you incurring a huge penalty). However, the solution should be reasonable enough. It shouldn't be too complicated too. For example, if you are asked to sort, we don't expect you to use merge sort. You can use a naive algorithm like bubble sort which solves the problem in $O(n^2)$. But if you come up with something overly complicated like a $O(n^3)$ or $O(n^4)$ solution and because of which your code fails to run even on small/simple test cases, then that can attract penalties.
- You must strictly stick to the **input and output formats**.
- Your C file can contain only inputs, outputs and memory allocations. Everything else should be in a function defined in assembly which will be invoked from C file.
- No need to handle invalid input cases unless explicitly asked to handle them in the question.
- No need to handle overflow unless explicitly mentioned in the question.

Submission format: Strictly adhere to the following submission format. Failure to do so may result in an erroneous evaluation of your exam. A minimum of -5 marks would be used as penalty if the submission instructions are not followed properly plus there is a good likelihood that your submission would not be evaluated if significant efforts are involved.

- The following directory structure is expected,

```
./<roll_number>
├── qa
│   ├── qa.s
│   └── qa.c
└── qb
    ├── qb.s
    └── qb.c
```

- For example, if you are assigned questions 2 and 8 and your roll number 2023101001 then the following submission is expected.

```
2023101001
├── q2
│   ├── q2.s
│   └── q2.c
└── q8
    ├── q8.s
    └── q8.c
```

Problem 1: Given a positive integer N, return an array of integers with all the integers from 1 to N. But for multiples of 2 the array should have -1 instead of the number, for multiples of 7 the array should have -2 instead of the number, and for multiples of both 2 and 7 the array should have -3 instead of the number.

Input/Output Format

- Input: N
- Output: N numbers from 1 to N with modifications as required

Sample Test Case

Input: 5

Output: 1 -1 3 -1 5

Input: 17

Output: 1 -1 3 -1 5 -1 -2 -1 9 -1 11 -1 13 -3 15 -1 17

Problem 2: Given a number N, check if it is a palindrome or not.? Palindromes are those numbers which read the same backward and forward. 1, 363, 1331 are palindromes while 10, 456 are not.

Input/Output Format

- Input: N, Single Integer to be checked
- Output: True, if it is a palindrome; False, if not.
- Note: Output case does not matter, TrUe, true, TRUE all are acceptable.

Sample Test Case

Input: 13931

Output: TRUE

Input: 69

Output: FALSE

Problem 3: You are given a string s. The **score** of a string is defined as the sum of the absolute difference between the **ASCII** values of adjacent characters.

Return the **score** of s.

$2 \leq s.length \leq 100$

s consists only of lowercase English letters.

Input/Output Format

- Input: String s.
- Output: Score of string s.

Sample Test Case

Input : s = "hello"

Output: 13

Problem 4: Given a 2-D array of non-negative integers, find the sum of all those integers which are divisible by 3 but not divisible by 5. You may initialize a static array (so that all the elements are contiguous, regardless of row).

You do not need to handle overflow. You can assume that the sum will fit in a long long integer, and that each number will fit in a long long integer.

Input/Output Format

- Input: M N, where M is the number of rows and N is the number of columns. Next M lines contain N integers each where ith line represents the elements of ith row of matrix.
- Output: Single integer which is the sum of the required numbers.

Sample Test Case

Input

3 3

1 2 3

4 9 0

5 6 105

Output

18

Problem 5: Given a sorted array of N integers where some elements may be repeated. The task is to process the input array and print only the unique elements present, in the same order.

Input/Output Format

- Input:
First line: Integer N ($1 \leq N \leq 10^5$) — the number of elements

Second line: N space-separated integers a_1, a_2, \dots, a_N ($-10^9 \leq a_i \leq 10^9$) in non-decreasing order.

- Output: Space-separated unique elements in sorted order

Sample Test Cases

Input:

6

1 1 3 5 6 6

Output:

1 3 5 6

Input:

4

1 2 3 4

Output:

1 2 3 4

Problem 6: Given an unsorted array of N integers, check whether the elements form a consecutive sequence of distinct integers (without any duplicates or missing values).

Input/Output Format

- Input:
The first line contains an integer N ($1 \leq N \leq 10^5$), the number of elements in the array.
The second line contains N space-separated integers a_1, a_2, \dots, a_N ($-10^9 \leq a_i \leq 10^9$)
- Output: Print "TRUE" if the array consists of consecutive numbers. Otherwise, print "FALSE".
- Note: Output case does not matter, TrUe, true, TRUE all are acceptable.

Sample Test Cases

Input:

6

5 3 2 1 4 6

Output:

TRUE

Input:

6

7 6 5 5 3 4

Output:

FALSE

Problem 7: Given a number N, check if the sum of the factorial of digits is equal to N (special number).

Input/Output Format

- Input: N
- Output: TRUE, if the input is special number, FALSE otherwise.
- Note: Output case does not matter, TrUe, true, TRUE all are acceptable.

Sample Test Case

Input : 145

Output: TRUE

Explanation- $1! + 4! + 5! = 145$

Problem 8: Given an array of N integers, sort the array into a wave-like array and return it. In other words, arrange the elements into a sequence such that $a_1 \geq a_2 \leq a_3 \geq a_4 \dots$

Note: If multiple answers are possible, return the lexicographically smallest one.

Input/Output Format

- Input: Has two lines. First line contains single integer N, size of the array. Next line contains N space-separated integers which are elements of the array.
- Output: Wave form of input array.

Sample Test Case

Input:

4

1 2 3 4

Output:

2 1 4 3

Input:

2

1 2

Output:

2 1

Problem 9: You are given a sorted array of integers and a search value. Your task is to write an assembly language program that implements the binary search algorithm using recursion to search for the given value in the array.

The program should implement a recursive binary search function to find the index of the search value in the array. If the value is found, the program should return the index. If the value is not found, the program should return -1. (0 based indexing should be followed for solution).

Input/Output Format

- Input: Contains three lines. First line contains a single value N , size of the array. ($1 \leq N \leq 50$)

The second line contains N distinct values in sorted order $a_1, a_2 \dots a_n$, elements of the array.

($-10^9 \leq a_i < 10^9$). The third line contains the search value

- Output: M , where M is the index of search element

Sample Test Case

Input:

5

1 3 5 7 12

7

Output:

3

Problem 10: Write an assembly language program to determine whether a given string of characters, consisting of parentheses (including '(', ')', and '['), is balanced or not. The program should use stack operations to solve the problem. The program should take an

input string and its length as arguments and return 1 if the parentheses in the string are balanced, and 0 otherwise. Using this returned value, make your C program output “BALANCED” if the string is balanced, and “NOT BALANCED” if the string is not balanced (do not print the quotes).

A balanced string is defined as one where each opening parenthesis has a corresponding closing parenthesis in the correct order. For example, the string “([{}])” is balanced, while the string “{[(())]” is not balanced. You can assume input string only contains parentheses.

You are required to implement a solution in $O(n)$ time complexity.

Input/Output Format

- **Input:** Contains two lines. The first line contains a single value N , which is the size of the string. ($1 \leq N \leq 10^8$)
- **Return Value:** 1 for balanced string or 0 for not balanced
- **Output:** BALANCED or NOT BALANCED
- **Note:** Output case does not matter, Balanced, balanced, BALANCED all are acceptable.

Sample Test Cases

Input

4

{() }

Output

BALANCED

Input

4

{() }

Output

NOT BALANCED

Problem 11: Given an array of integers as input (you need to take this as input in your C code), find the nearest previous greater integer (i.e., nearest greater integer to the left) for each of the integers in the array. In case no such integer exists for any of the integers, the answer is -1 . Write a function in x86 assembly code for the problem.

You are required to implement a solution in $O(n)$ time complexity.

Input/Output Format

- Input: Contains two lines. First line contains a single value N , size of the array. ($1 \leq N \leq 10^8$). Second line contains N space-separated integers.
- Output: Sequence of N integers, b_1, b_2, \dots, b_N , where b_i ($1 \leq i \leq N$) is the integer that's the nearest previous greater integer (to the left) of the i th element.

Sample Test Case

Input:

6

4 6 5 2 10 8

Output:

-1 -1 6 5 -1 10

Problem 12: Given a number N and the number of digits D in it, check whether the given number is a circular prime. A circular prime is a number that remains prime under all rotations of its digits. A rotation is defined as moving the last digit to the front.

Input/Output Format

- Input: A single line containing two space separated integers N and D , ($0 < N < 10^5$)
- Output: Print "TRUE" if N is a circular prime. Otherwise, print "FALSE".
- Note: Output case does not matter, TrUe, true, TRUE all are acceptable.

Sample Test Cases

Input: 197 3

Output: TRUE

Input: 1193 4

Output: FALSE

ALL THE BEST!