INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSC-101 PROGRAMMING WITH C AND C++

Autumn 2023 Problem Set 3

---

1. (Tic-Tac-Toe Checker) Write a C program to check the status of a Tic-Tac-Toe board. The program should be able to recognize if either player has won, or if the game is still ongoing, or it's a draw.

   Input:

   The program should take a 3x3 matrix of characters as input where 'X' represents one player, 'O' represents the other player, and '-' represents an empty spot on the board.

   For example:

   ```
   X O X
   O X O
   X - -
   ```

   Output:

   The program should output one of the following statuses:

   - "Player X Wins!"

   - "Player O Wins!"

   - "It's a Draw!"

   - "Game still ongoing"

   Example:

   For the board shown above, the output should be:

   Player X Wins!

2. (Minimum Distance Finder) Write a program that finds two elements in an array such that the absolute difference between them is the smallest among all possible pairs of elements in the array.

   Input:

   The first line contains an integer N ($2 \leq N \leq 1000$), representing the number of elements in the array. The next line contains N distinct integers, a_i ($-10^6 \leq a_i \leq 10^6$), separated by spaces, representing the elements of the array.

Output:

Output two integers, the pair of elements with the minimum absolute difference. If there are multiple such pairs, output the one that appears first.

Example:

Input:

5

3 7 0 8 -2

Output:

7 8

3. (Perfect Square Pairs) You need to write a program that takes an array of integers and finds all pairs of numbers whose product is a perfect square. These pairs should be distinct and ordered in the sequence they appear in the array.

   Input:

   - The first line contains an integer 'N' ($1 \leq N \leq 1000$), the size of the array.
   - The second line contains 'N' integers separated by spaces.

   Output:

   For each pair of numbers whose product is a perfect square, print them on a new line in the order they appear in the array. If there are no such pairs, output "No perfect square pairs found".

   Example:

   Input:

   6

   2 4 8 16 1 3

   Output:

   2 8

   4 16

   ...

   Explanation:

   - '2 * 8 = 16' which is a perfect square (4^2)

   - '4 * 16 = 64' which is also a perfect square (8^2)

4. (Rotated Palindromes) A "Rotated Palindrome" is a string that can become a palindrome if it is rotated (circularly shifted to the left) one or more times. Write a program to check if a given string is a Rotated Palindrome or not.

   Input:

   - The input consists of a single line containing a non-empty string made up of lowercase English letters. The string will have at least 1 and at most 10^5 characters.

   Output:

   - Print "YES" if the given string is a Rotated Palindrome. Otherwise, print "NO".

   Example:

   Input:

   babbbbb

   Output:

   YES

   Explanation:

   The string "babbbbb" can be rotated to "bbabbbb" which can be further rotated to "bbbabbb", which is a palindrome.

5. (Uncommon Words Histogram) A paragraph of text is given, with words separated by spaces. Write a program to find the frequency of each word that appears only once in the paragraph and display the word along with its frequency. You need to ignore case sensitivity, and consider words that are the same but have different cases as identical.

   Input:

   The first line contains a string 'S' representing the paragraph ($1 \leq |S| \leq 1000$).

   Output:

   Print all words that appear only once in the paragraph, along with their frequency, in the order they first appear. Each word and its frequency should be printed on a new line. If no such words exist, print "No uncommon words found".

   Example:

   Input:

   Programming is fun. It's a creative process and the possibilities are endless.

   Output:

   Programming 1

   is 1

fun. 1

It's 1

a 1

creative 1

process 1

and 1

the 1

possibilities 1

are 1

endless. 1

6. (Matrix Pattern Recognition) Given a 2D array (matrix) of integers, your task is to write a program to find if there is any row or column which has all the same values.

   Input:

   - The first line contains an integer n ($2 \leq n \leq 100$), the number of rows of the matrix.
   - The second line contains an integer m ($2 \leq m \leq 100$), the number of columns of the matrix.
   - The next n lines, each line contains m integers (separated by space) representing the values in the matrix.

   Output:

   Print "YES" if there is at least one row or one column where all values are the same. Otherwise, print "NO". If there are multiple valid rows or columns, just printing "YES" for one of them is enough.

   Example:

   Input:

   ```
   3
   4
   1 2 2 2
   2 2 2 2
   3 3 3 3
   ```

   Output:

   YES

7. (Alphabetic Encryption) You are tasked with creating a simple encryption algorithm based on alphabets. Here's the algorithm:

   - The English alphabet is divided into two parts: vowels (A, E, I, O, U) and consonants (the rest).
   - Each vowel is assigned a prime number (A=2, E=3, I=5, O=7, U=11).
   - Each consonant is assigned a number based on its position in the alphabet, omitting the vowels (B=1, C=2, . . . , Y=20, Z=21).
   - Your task is to write a program that takes a word as input and outputs the encrypted number which is the sum of the assigned numbers to each character of the word. Case of the letters should not affect the outcome.

   Input:

   The input consists of a single non-empty word in uppercase or lowercase, and the word does not contain any other characters. The length of the word is at most 100 characters.

   Output:

   Output the encrypted number as described above.

   Example:

   Input:

   Hello

   Output:

   34

   Explanation:

   H=8, E=3, L=12, L=12, O=7

   6 + 3 + 9 + 9 + 7 = 34

8. (Localized Sorting) You are given an array of integers. Your task is to sort all odd numbers in ascending order while leaving the even numbers at their original positions.

   Input:

   - The first line contains an integer 'N' ($1 \leq N \leq 100$), representing the size of the array.
   - The next line contains 'N' space-separated integers.

Output:

Print the modified array where all odd numbers are sorted in ascending order, and even numbers are in their original positions.

Example:

Input:

10

5 3 2 8 1 4 11 10 9 7

Output:

1 3 2 8 5 4 7 10 9 11

9. (Smart Subsequences) Write a C program that finds and prints all subsequences of an array of integers where each subsequence:

   - Is of length at least 2.
   - Its elements are in strictly increasing order.
   - The sum of its elements is a perfect square.

   Input:

   - The first line contains an integer N ($2 \leq N \leq 100$), the size of the array.

   - The next line contains N integers, $a_1, a_2, \ldots, a_N$ ($-10^6 \leq a_i \leq 10^6$), separated by spaces.

   Output:

   - For each valid subsequence, print its length followed by its elements, each separated by a space. Each subsequence should be printed on a new line.

   - If there are multiple valid subsequences, they can be printed in any order.

   - If no valid subsequences are found, print "No valid subsequences found".

   Example:

   Input:

   5

   1 2 3 5 8

   Output:

   2 1 8

   3 1 3 5

   Explanation:

- The subsequences [1, 8] and [1, 3, 5] are strictly increasing and their sums (9 and 9 respectively) are perfect squares.

10. (Social Network Connection Scores) In a simplified social network, each user has a connection score with every other user, based on their interactions. This score is represented as an integer. Your task is to write a program to find and display the pairs of users with the highest connection score in the network.

    Input:

    - An array 'scores[N][N]' where 'N' is the number of users ($2 \leq$ N ≤ 100) and 'scores[i][j]' is the connection score between user 'i' and user 'j' ($0 \leq$ scores[i][j] ≤ 1000). It is given that 'scores[i][j]' is equal to 'scores[j][i]' and 'scores[i][i]' is always 0.
    - The scores are symmetric and the diagonal elements are zero.

    Output:

    - Print the pairs of users (user indices) with the highest connection score. If multiple pairs have the same highest score, print all such pairs.

    Example:

    Consider an example where 'N=4' and the array 'scores' is as follows:

    ```
    0   10 20 30
    10   0 40 50
    20 40   0 60
    30 50 60   0
    ```

    Output:

    2 3

    Instructions:

    1. Read the value of 'N' and the 'N x N' array 'scores' from the user.

    2. Process the array to find the pair(s) with the highest connection score.

    3. Print the result as specified in the output section.

    Bonus Challenge:

    Optimize your solution in terms of time complexity and memory usage.