

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSC-101 PROGRAMMING WITH C AND C++

Autumn 2023

Problem Set 2

1. (Sequence Modifier) Given an array of integers, your task is to implement a program that does the following:
 - Determine if the current number is prime or not.
 - If the number is prime, increase the number by the total count of prime numbers found so far in the array.
 - If the number is not prime, decrease the number by its highest prime factor.
 - Print the modified array at the end.

Input:

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

Output:

Print the modified array.

Example:

Input:

5

2 15 9 13 7

Output:

3 10 6 16 9

2. (Pythagorean Triples) A Pythagorean triple consists of three positive integers a , b , and c , such that $a^2 + b^2 = c^2$. You are given an array of integers. Your task is to write a program to check if the array contains any three numbers which can form a Pythagorean triple. If the array contains such numbers, print them, else print "No Pythagorean triples found".

Input:

The first line contains an integer 'N' ($3 \leq N \leq 50$), the size of the array. The next line contains 'N' positive integers separated by spaces.

Output:

If a Pythagorean triple exists in the array, print the numbers in increasing order separated by spaces. If multiple triples exist, print all of them in separate lines. If no triple exists, print "No Pythagorean triples found".

Example:

Input:

6

5 12 13 15 9 8

Output:

5 12 13

Bonus Challenge: Can you find a solution that is efficient in terms of time complexity? An obvious approach is to try every combination, but think if there is a faster way!

3. (Prime Pals) A “Prime Pal” is a pair of prime numbers in an array such that the absolute difference between them is less than 10, and they are adjacent to each other in the array. Write a program to locate all Prime Pals in a given array of integers.

Input:

The first line contains an integer ‘N’ ($2 \leq N \leq 100$), the size of the array. The next line contains ‘N’ integers separated by spaces.

Output:

For each Prime Pal in the array, print the pair in the order they appear in the array, separated by a space. Each pair should be printed on a new line. If no such pairs exist, print “No Prime Pals found”.

Example:

Input:

7

3 5 15 29 31 44 7

Output:

3 5

29 31

4. (Consonant Sandwich) A “Consonant Sandwich” is a string that starts and ends with a vowel, but has only consonants in between. Write a program that takes a string as input and determines if it’s a “Consonant Sandwich”. Furthermore, if it is a “Consonant Sandwich”, your program should also print the number of consonants between the starting and ending vowel.

Input:

A single string ‘S’ ($3 \leq |S| \leq 100$) containing only lowercase alphabets.

Output:

If the string is a “Consonant Sandwich”, print “Yes” followed by the number of consonants between the first and last vowel. If not, print “No”.

Example:

Input:

ultrm

Output:

Yes 3

5. (Magic Sequences) A sequence of integers is said to be “magical” if the absolute difference between any two consecutive numbers is always 1, and the sequence never dips below 1. Your task is to:

- Accept a number ‘N’ from the user, indicating the length of the sequence.
- Accept ‘N’ integers representing the sequence.
- Determine if the given sequence is magical or not.
- If it is, print the maximum number achieved in the sequence.
- If it’s not, print which number (or numbers, if multiple) break the sequence.

Input:

- An integer ‘N’ ($2 \leq N \leq 100$) - the length of the sequence.

- ‘N’ integers ($1 \leq \text{integer} \leq 100$).

Output:

If the sequence is magical, print:

Magical: [Maximum number in sequence]

If the sequence isn’t magical, print:

Not Magical: [Number(s) that break the sequence]

Example:

Input:

5

1 2 3 2 1

Output:

Magical: 3

Input:

6

1 2 3 5 2 1

Output:

Not Magical: 5

6. (Palindrome Reversals) A palindrome is a word, phrase, number, or other sequences of characters that reads the same forward and backward (ignoring spaces, punctuation, and capitalization). For example, “radar” and “A man, a plan, a canal, Panama!” are palindromes. Your challenge:

- Take a string input 'S' from the user.
- Determine the minimum number of characters you need to reverse (from the end to the beginning) to make the string a palindrome.
- If the string is already a palindrome, print a suitable message.
- Display the new palindrome.

Input:

- A string 'S' ($1 \leq \text{length of } S \leq 100$).

Output:

- If the string is already a palindrome, print:

Already a palindrome!

- If the string needs reversal to become a palindrome, print:

Reversed [number of characters reversed] characters.

New Palindrome: [The newly formed palindrome]

Example:

Input:

openai

Output:

Reversed 5 characters.

New Palindrome: openaianepo

Input:

madam

Output:

Already a palindrome!

7. (Spiral Square) Imagine a square spiral which starts with the number '1' and moves to the right. Each step in the spiral increments the current number by 1. The pattern then turns up, left, down, right, and so on, forming a square spiral. Consider the following code that generates this spiral:

```

1 #include<stdio.h>
2 int main()
3 {
4     int n = 5;
5     int spiral[n][n];
6     int top = 0, bottom = n-1, left = 0, right = n-1;
7     int num = 1, dir = 0;
8
9     while (top <= bottom && left <= right) {
10         if (dir == 0) {

```

```

11         for (int i = left; i <= right; i++)
12             spiral[top][i] = num++;
13         top++;
14         dir = 1;
15     } else if (dir == 1) {
16         for (int i = top; i <= bottom; i++)
17             spiral[i][right] = num++;
18         right--;
19         dir = 2;
20     } else if (dir == 2) {
21         for (int i = right; i >= left; i--)
22             spiral[bottom][i] = num++;
23         bottom--;
24         dir = 3;
25     } else if (dir == 3) {
26         for (int i = bottom; i >= top; i--)
27             spiral[i][left] = num++;
28         left++;
29         dir = 0;
30     }
31 }
32 for (int i = 0; i < n; i++) {
33     for (int j = 0; j < n; j++) {
34         printf("%d ", spiral[i][j]);
35     }
36     printf("\n");
37 }
38 return 0;
39 }

```

Given ‘n’, this code prints the first ‘n x n’ numbers in a square spiral. For ‘n = 3’, the output will be:

```

1 2 3
8 9 4
7 6 5

```

Based on the output sequence for ‘n = 4’, if $(c_k)_1^P$ represents the sequence traversed column-wise (i.e., first all entries of the first column, then all of the second column, and so on), find the value of:

$$\frac{c_2 \times c_{P-1}}{c_{\lceil \frac{P}{3} \rceil}}$$

Where $\lceil . \rceil$ denotes the smallest integer not less than the enclosed value (ceiling function).