INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSC-101 PROGRAMMING WITH C AND C++

Autumn 2023 Problem Set

---

1. (Library) Consider a library that has n books available for rent. Each book has a popularity index in units as $p_1, p_2, p_3, p_4, \ldots, p_n$. Suppose a member is given a special pass which allows him to rent a subset of books $= K < n$ for a month. Write the pseudocode for the program that will determine the maximum possible total popularity index for the subset of books he/she can choose.

2. (Equations) Given a system of equations:

$$a^2 + b = n$$
$$a + b^2 = m$$

   Find the count of pairs of integers (a, b), such that $0 \leq a, b$, which satisfy this system.

   Input:

   A single line that contains two integers $n, m (1 \leq n, m \leq 1000)$. These are the parameters of the system. The numbers on the line are separated by a space.

   Output:

   Print the answer to the problem on a single line.

3. (UID) Suppose there's a stack of books in a library. Each book has a unique identification number, and the librarian marks the same number on a card given to the person who borrows the book. The stack of books have the following sequence of numbers: 101, 102, 103, 104, 105, 106, 107, 108, 109, 110 and are stacked in the order mentioned. One day, a mischievous student decides to reverse the order of a portion of the sequence of books. After his mischief, the stack looks like this: 108, 107, 106, 105, 104, 103, 102, 101, 109, 110. Write an efficient C program to find the order of books after reversing a portion of the stack.

4. (Subsequences) Given two integer arrays $a_1, \ldots, a_n$ and $b_1, \ldots, b_m$, your task is to identify a non-empty array $c_1, \ldots, c_k$ that represents a subsequence present in both original arrays. The target is to find the shortest possible subsequence, in the event of multiple sequences with the same shortest length, you may select any. If no common subsequences exist, this should be reported.

   A sequence a is a subsequence of a sequence b if sequence a can be derived by deleting some elements (including possibly none) from b. For instance, [3,1] is a subsequence of [3,2,1] and [4,3,1], but is not a subsequence of [1,3,3,7] or [3,10,4].

Input:

The initial line contains an integer $t (1 \leq t \leq 1000)$ representing the number of test cases. Following this are $3t$ lines describing the test cases. Each test case starts with a line containing two integers n and $m (1 \leq n, m \leq 1000)$, the lengths of the two arrays. The subsequent line of each test case contains n integers $a_1, \ldots, a_n (1 \leq a_i \leq 1000)$ representing the elements of the first array. The third line of each test case contains m integers $b_1, \ldots, b_m (1 \leq b_i \leq 1000)$ representing the elements of the second array. It's guaranteed that the total of n and m across all test cases won't exceed 1000 $(\sum_{i=1}^{t} n_i, \sum_{i=1}^{t} m_i \leq 1000)$.

Output:

For each test case, output "YES" if a common subsequence exists, or "NO" if not. If the answer is "YES", then on the next line output an integer $k (1 \leq k \leq 1000)$ indicating the length of the common subsequence, followed by k integers $c_1, \ldots, c_k (1 \leq c_i \leq 1000)$ which are the elements of the common subsequence. In cases where multiple smallest k solutions exist, output any.

Example:

```
Input:
5
4 5
10 8 6 4
1 2 3 4 5
1 1
3
3
1 1
3
2
5 3
1000 2 2 2 3
3 1 5
5 5
1 2 3 4 5
1 2 3 4 5

Output:
YES
1 4
YES
1 3
NO
YES
1 3
YES
1 2
```

Note:

In the first scenario, [4] is a common subsequence of [10,8,6,4] and [1,2,3,4,5]. The length of this subsequence is 1, which is the shortest possible length for a common subsequence of both a and b.

In the third scenario, no common subsequences between [3] and [2] exist, hence the output is "NO".

5. (Geometry) There is a chip located at point (0,0) on a coordinate plane. In one move, you can transfer the chip from one point (x_1, y_1) to another point (x_2, y_2) if the Euclidean distance between these two points is an integer (i.e., the value of $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ is an integer). The challenge is to figure out the least number of moves necessary to relocate the chip from the origin point (0,0) to a target point (x, y).

   Input:

   The initial line contains a single integer $t (1 \leq t \leq 3000)$, representing the number of test cases. Each test case comprises a single line with two integers, x and $y (0 \leq x, y \leq 50)$, specifying the coordinates of the destination point.

   Output:

   For each test case, print a single integer, denoting the minimum number of moves necessary to transport the chip from the origin (0,0) to the destination point (x, y).

   Example:

```
Input:
3
8 6
0 0
9 15

Output:
1
0
2
```

   Note:

   In the first test, just one move from $(0, 0)$ to $(8, 6)$ would be enough, as the Euclidean distance $\sqrt{(0-8)^2 + (0-6)^2} = \sqrt{64+36} = \sqrt{100} = 10$ is an integer. In the second test, the chip is already at the target location. In the third test, the chip can be transported in two moves: first, from $(0, 0)$ to $(5, 12)$, and then from $(5, 12)$ to $(9, 15)$. The Euclidean distances $\sqrt{(0-5)^2 + (0-12)^2} = \sqrt{25+144} = \sqrt{169} = 13$ and $\sqrt{(5-9)^2 + (12-15)^2} = \sqrt{16+9} = \sqrt{25} = 5$ are integers.

6. (Launch) Analyze the following code snippet:

```c
#include<stdio.h>
int main()
{
    int j = 5;
    do
    {
        printf("%d", j);
        printf("Launching the mission. ");
    } while(printf("Countdown ") && (--j > 2));
    return 0;
}
```

If p represents the number of punctuation marks in the output of the above code, and n denotes the number of numerals, then report the value of $\left\lceil \frac{n}{p} \right\rceil$? Here, $\lceil . \rceil$ denotes the least integer function.

7. (Sequence) Consider the following code:

```c
#include<stdio.h>
int main()
{
    int x = 2, y = 1;
    do {
        printf("%d*%d ", x, y);
        x += 2;
        y *= x;
    } while (x <= 8);
    return 0;
}
```

Given the output sequence of the code as $(b_n)_1^M$, for each term being represented by $x * y$ at that instance, what will be the value of:

$$\frac{b_1 \times b_M}{b_{\lfloor \frac{M}{2} \rfloor + 1}}$$

Where, $\lfloor . \rfloor$ denotes the greatest integer function.