Problem Description

Problem Statement

For this assignment, you are asked to implement an algorithm to solve Problem 2 of Assignment 6. Let us recall the problem:

You are given an array of $n$ distinct natural numbers $A[1..n]$, each containing $k$ digits, where the most significant digit is in $\{1..9\}$. Your goal is to find an optimal path from $A[1]$ to $A[n]$. What do we mean by a path? You can move from $A[i]$ to $A[j]$ if they differ in one digit, and you may only move from numbers in $A$ to other numbers in $A$.

So, for example, if $A = [1234, 4321, 3234, 5274, 3274, 5074]$, you cannot move directly from 1234 to 5074, because they differ in three digits: $1 \neq 5$, $2 \neq 0$, $4 \neq 7$. Similarly, you cannot move from 1234 to 4321 since they differ in four digits. However, there is a path from $A[1]$ to $A[n]$, namely, 1234 to 3234 to 3274 to 5274.

What do we mean by optimal? Moving from $A[i]$ to $A[j]$ costs $2A[j] - A[i]$. An optimal path has the smallest total cost. Note that the cost might actually be negative (representing profit). In the example above, the stated path is also optimal.

Write a program to find any optimal path from $A[1]$ to $A[n]$, or report that there is no such path.

Input Format

The first line of input consists of $n$ and $k$, separated by one space.

The second line of input consists of $A[1], A[2], \ldots, A[n]$, each separated by one space.

Output Format

If there is no path from $A[1]$ to $A[n]$, output IMPOSSIBLE.

Otherwise, output two lines. On the first line, output two integers $C$ and $L$, where $C$ is the optimal cost and $L$ is the number of vertices on the optimal path you find. On the next line, output $p_1, p_2, \ldots, p_L$, the optimal path you find. The path should have cost $C$ and satisfy:

- each $p_i$ is in $A$;
- $p_1 = A[1], p_L = A[n]$;
- $p_i$ and $p_{i+1}$ differ in one digit, for $1 \leq i < L$. 


Note: The checker is case-sensitive, so please output IMPOSSIBLE in all caps. Separate integers on the same line by one space. End each line with no trailing spaces and one newline character.

Constraints
For all test cases, $2 \leq n \leq 1000$, $1 \leq k \leq 8$, and the $A[i]$'s are distinct $k$-digit natural numbers (i.e. $10^{k-1} \leq A[i] < 10^k$, and $A[i] \neq A[j]$ if $i \neq j$).

Sample Input 1
6 4
1234 4321 3234 5274 3274 5074

Sample Output 1
20696 5
1234 3234 3274 5274 5074

Sample Input 2
7 3
100 900 990 199 130 139 999

Sample Output 2
2366 5
100 130 139 199 999

Sample Input 3
2 2
67 76

Sample Output 3
IMPOSSIBLE

Explanation
Sample I/O 1 is the same as the example given in the problem statement. The total cost is $2 \cdot 3234 - 1234 + (2 \cdot 3274 - 3234) + (2 \cdot 5274 - 3274) + (2 \cdot 5074 - 5274) = 20696$.
Sample I/O 2 shows that the optimal path may not be a shortest path.
Submission Instructions

- Submit your solution on Marmoset.
- You can choose to code in either C++ or Python.
- Name your program prog2.cpp/prog2.py.
- **Time limit**: 3 seconds (C++) / 12 seconds (Python) for each test case.
- Compilation command for C++: g++ -std=c++14 prog2.cpp -O3 -o prog2.
- Execution command for Python: python3 prog2.py.
- Read from standard input and write to standard output.
- There will be several test cases, worth a total of 30 points. The public tests are worth 8 points and the secret tests are worth 22 points. The public tests (input only, not the answer) will be made available under a separate file.
- We will take the submission with the highest score. Please, however, refrain from excessive submissions.
- General collaboration policy applies. Please acknowledge your collaborator(s) by adding a comment in the beginning of your code.
- FAQ and updates will be posted on Piazza when necessary.

Hints

- There are (at least) two possible ways to represent the input elements \(A[i]\): as integers, or as strings. It may not be necessary, but if you wish to convert between the two types, in C++ you can use `stoi()`/`stol()`, `strtol()` and `to_string()` (or C-style functions `atoi()`/`atol()`, `sscanf()` and `sprintf()`), and in Python you can use `int()` and `str()`.
- For C++ users, 64-bit integer type may be needed.