"every challenge is an opportunity to grow

and

Every problem you solve is a step closer to your goals"

Contents

A. Gift Carpet

Recently, Mika and Vika celebrated Family Day. Their friend Emily gave them a carpet, which can be represented as an $n \cdot m$ table of lowercase Latin letters.

Vika hasn't seen the gift yet, but Mika knows what kind of carpets she likes. Vika will like the carpet if she can read her name on it. She reads column by column from left to right and chooses one or zero letters from the current column.

Formally, the girl will like the carpet if it is possible to select four distinct columns in order from left to right such that the first column contains "v", the second one contains "i", the third one contains "k", and the fourth one contains "a".

Help Mika understand in advance whether Vika will like Emily's gift.

Input format:

Each test consists of multiple test cases. The first line of input contains a single integer t ($1 \le t \le 100$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains two integers n, m ($1 \le n,m \le 20$) — the sizes of the carpet.

The next n lines contain m lowercase Latin letters each, describing the given carpet.

Output format:

For each set of input data, output exactly **"YES"** if Vika likes the carpet, otherwise output exactly **"NO"**.

Time Limit: 1 sec

Example on the next page.

Input:

```
4
1 4
vika
3 3
bad
car
pet
4 4
vvvv
iiii
kkkk
aaaa
4 4
vkak
iiai
avvk
viaa
```

Output:

```
YES
NO
YES
NO
```

B. Prime Factorization

Given an integer, your task is to find its prime factorization.
Prime factorization is the process of finding the prime numbers that multiply together to create a given integer.

Input format:

The input consists of multiple test cases. Each test case starts with an integer n (1 <= n<=10^6), representing the number for which you need to find the prime factorization.
It is guaranteed that the input is always a valid number.

Output format:

For each test case, print the prime factorization of the number as a list of prime factors. The list should be in ascending order.

Time Limit: 3 seconds

Example:

Input:

3
12
36
105

Output:

2 2 3
2 2 3 3
3 5 7

C. Ancient Treasure Hunt

In a faraway land, there lies a map that holds the key to an ancient treasure. However, the map is not an ordinary one; it's encoded in two scrolls, each filled with a mystical sequence. To uncover the map's secrets, you must find the length of the longest shared path between these scrolls.

Input format:

The input consists of multiple test cases. Each test case represents a pair of scrolls. The scrolls are unique and ancient, with distinct inscriptions. Your task is to find the length of the longest shared path between them.

Output format:

For each test case, print the length of the longest shared path between the scrolls.

Constraints:

The scrolls' inscriptions have lengths between 1 and 1000 characters.
The inscriptions consist of ancient symbols.
The inscriptions are unique for each test case.

Time Limit: 2 Secs

Examples:

Test Case 1:

Input:

2
Scroll1
Scroll2
Ancient
Inscriptions

Output:

6
4

6

D. Subset Sum

Given an array of integers and a target sum, find if there exists a subset of the array elements that sums up to the target.

Input format:

The input consists of multiple test cases. Each test case starts with an integer n (1 <= n <= 20), representing the number of elements in the array. The next line contains n integers representing the array elements. The last line contains an integer target (-1000 <= target <= 1000).

Output format:

For each test case, print "Yes" if there exists a subset of the array elements that sums up to the target; otherwise, print "No."

Time Limit: 2 secs **(Be careful about the time limit for this problem!. A brute force solution will exceed the time limit)**

Example:

Input:
2
4
1 2 3 4
5
3
7 15 10
13

Output:
Yes
No

E. Where to Meet in Gridland

In Gridland (Matrix World) there are some friends living. one day they decided to meet each other, but they didn't know where is exactly the best place to meet so each friend travels a fair distance.

Help them find the best palace where the total distance travel is minimal.

Given a binary grid where each 1 represents a friend's residence, return the shortest possible **total travel distance**.

The **total travel distance** is the sum of the distances between the houses of the friends and the meeting point. Using the Manhattan Distance formula, the distance is determined as follows: distance(p1, p2) = [p2.x - p1.x] + [p2.y - p1.y].

Input format:

The first line of input will contain a single integer T, denoting the number of test cases. Each test case will have two space-separated integers M and N denoting the number of rows and columns respectively and then there will be the binary matrix.

Output format:

For each test case, output a single integer, denoting the minimum travel distance

Time limit: 3 secs

Constraints:

- $1 \leq T \leq 100$
- 1 <= M, N <= 200
- The grid will have a minimum of two friends. Example

Example on the next page

Input:
2
3 5
1 0 0 0 1
0 0 0 0 0
0 0 1 0 0
1 2
1 1

Output:
6
1

Expalin:
In the first test case you have three friends who live in (0,0), (0,4), and (2,2). The position (0,2) makes for an optimal meeting place, since the total travel distance of 2 + 2 + 2 = 6 is the shortest, hence, print 6.

F. The Permutaion Problem

Andrey is just starting to come up with problems, and it's difficult for him. That's why he came up with a strange problem about permutations† and asks you to solve it. Can you do it?

Let's call the cost of a permutation p of length n the value of the expression:

$$(\sum_{i=1}^{n} p_i \cdot i) - (\max_{j=1}^{n} p_j \cdot j)$$

Find the maximum cost among all permutations of length n.

A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, [2,3,1,5,4]is a permutation, but 1,2,2] is not a permutation (2 appears twice in the array), and [1,3,4] is also not a permutation (n=3 but there is 4 in the array).

Input format:

Each test consists of multiple test cases. The first line contains a single integer t (1≤t≤30) — the number of test cases. The description of the test cases follows.
The only line of each test case contains a single integer n (2≤n≤250) — the length of the permutation.It is guaranteed that the sum of n over all test cases does not exceed 500.

Output format:

For each test case, output a single integer — the maximum cost among all permutations of length n.

Time limit: 3 seconds

Example on the next page.

Input:
5
2
4
3
10
20

Output:
2
17
7
303
2529

Explain:

In the first test case, the permutation with the maximum cost is [2,1]. The cost is equal to $2\cdot 1+1\cdot 2-\max(2\cdot 1,1\cdot 2)=2+2-2=2$.

In the second test case, the permutation with the maximum cost is [1,2,4,3]. The cost is equal to $1\cdot 1+2\cdot 2+4\cdot 3+3\cdot 4-4\cdot 3=17$.