AIPO 2025
Preliminary Round

December 23, 2024

Contributors

The AIPO organisers would like to thank the following people.

For leading the question writing:

- Dr Sabin Tabrica
- Andrew Nash
- Ştefan-Bogdan Marcu
- Yanlin Mi
- Fangda Zou

Time limits and input/output bounds will become available when the submission system opens.

1 Add Two Numbers

You are given 2 integers a, b as standard inputs to your program. Print the sum a+b on a single line. A tutorial for this problem will be available on our website https://aipo.ucc.ie. This problem is intended to be a simple test to make sure you are able to understand the operations of the submission server.

Input/Output

Input

Two space-separated integers a, b on a single line.

Output

A single integer on a single line

Constraints

$-1,000,000 \leq a, b \leq 1,000,000$

Examples

Sample Input 1

```
3 2
```

Sample Output 1

```
5
```

Sample Input 2

```
-5 5
```

Sample Output 2

```
0
```

Sample Input 3

```
1000 -54391
```

Sample Output 3

```
53391
```

2 Close Numbers

Two integers n, m are called "close numbers" if:

1. n and m have the same number of digits
2. n differs from m by one digit
3. the digit of n is either greater or smaller by 1 than the digit of m.

For example, if $n = 1234$ and $m = 1224$ are close numbers as they have 4 digits and they have only the third digits different by 1. If $n = 1234$ then this will have lots of close numbers, e.g. 2234, 1134, 1334, 1224, 1244, 1233 and 1235 however, 0234 will not be a close number as 0234 is actually 234.

This problem requires finding how many close numbers an integer input n will have.

Input/Output

Input

One line containing the integer n.

Output

One integer, representing the number of close numbers that n has.

Constraints

- $0 \leq n \leq 1\,000\,000\,000$

Examples

Sample Input 1

```
1903
```

Sample Output 1

```
5
```

Explanation 1

Close Numbers: 1803, 1902, 1904, 1913, 2903

Sample Input 2

```
1234
```

Sample Output 2

```
7
```

Explanation 2

Close Numbers: 2234, 1134, 1334, 1224, 1244, 1233, 1235

3 Earthquake

The main goal of the Irish National Seismic Network (INSN) operation is to detect and locate earthquakes in and near Ireland using a network of seismometers. One seismometer transmits a sequence of 0 and 1 to the INSN base station every second for the duration of monitoring. Because 0 means no seismic activity while 1 senses an earth movement, an earthquake is identified as a sequence of 1-s that follows at least 2 0-s and is followed by at least 2 0-s. For example, 01001111011 does not show any earthquake, while 010011110011 identifies one earthquake of length 4.

Given a sequence of 0-s and 1-s from one seismometer, you are required to write a program to find the maximum duration of an earthquake expressed in seconds.

Input/Output

Input

1. The first line contains an integer n representing the length of the seismometer sequence.
2. The second line gives a sequence of length n containing 0-s and 1-s separated by a space.

Output

1. One integer representing the maximum duration in seconds

Constraints

- $5 \leq n \leq 100\,000$
- $1 \leq$ earthquake length $\leq n - 4$

Examples

Sample Input 1

```
11
01001111011
```

Sample Output 1

```
0
```

Sample Input 2

```
11
01001110011
```

Sample Output 2

```
3
```

Sample Input 2

```
21
001111000001010011001
```

Sample Output 2

```
4
```

4 Shipyard

A cargo ship has docked at a bustling shipyard in Cork, and its contents need to be transported to a nearby warehouse. This ship contains a line of n containers, each labelled sequentially from 1 to n. Lorries stand by to carry the containers and empty the ship. Each lorry can only carry a maximum of m containers at a time in one trip.

The containers are transported in order. A lorry can pick up to m containers in one go. For instance, if a lorry begins with container 3, it can transport containers $3, 4, \cdots, 3 + m - 1$, provided it does not exceed container n. The lorries do not have to be full in order to transport.

Your task is to determine the total number of ways the containers can be transported to the warehouse such that all containers are successfully moved. The total number presented to the output should be the total number of ways modulo (%) %20 242 024.

Input/Output

Input

1. Two integers n and m separated by a space. n - the number of containers in the cargo ship, m the maximum number of containers that a lorry can transport

Output

1. One integer r representing the total number %20 242 024 of ways in which the cargo ship can be emptied.

Constraints

- $m \leq n$
- for at least 30% of test cases $n \leq 1\,000$ and $m = 2$
- for an additional 30% of test cases $n \leq 1\,000$ and $m \leq 100$
- for the last 40% $n \leq 100\,000$ and $m \leq 10\,000$

Examples

Sample Input 1

```
4 2
```

Sample Output 1

```
5
```

Explanation

With 4 containers and lorries that can carry up to 2 containers, the possible ways to transport the containers are:

1. $[1, 2] \rightarrow [3, 4]$
2. $[1, 2] \rightarrow [3] \rightarrow [4]$
3. $[1] \rightarrow [2, 3] \rightarrow [4]$
4. $[1] \rightarrow [2] \rightarrow [3, 4]$
5. $[1] \rightarrow [2] \rightarrow [3] \rightarrow [4]$

Sample Input 2

```
5 3
```

Sample Output 2

```
13
```

Explanation

With 5 containers and lorries that can carry up to 3 containers, the possible ways to transport the containers are:

1. $[1, 2, 3] \rightarrow [4, 5]$
2. $[1] \rightarrow [2, 3, 4] \rightarrow [5]$
3. $[1, 2] \rightarrow [3, 4, 5]$
4. $[1, 2] \rightarrow [3, 4] \rightarrow [5]$
5. $[1, 2] \rightarrow [3] \rightarrow [4, 5]$
6. $[1, 2, 3] \rightarrow [4] \rightarrow [5]$
7. $[1] \rightarrow [2, 3] \rightarrow [4, 5]$
8. $[1] \rightarrow [2] \rightarrow [3, 4, 5]$
9. $[1, 2] \rightarrow [3] \rightarrow [4] \rightarrow [5]$
10. $[1] \rightarrow [2, 3] \rightarrow [4] \rightarrow [5]$
11. $[1] \rightarrow [2] \rightarrow [3, 4] \rightarrow [5]$
12. $[1] \rightarrow [2] \rightarrow [3] \rightarrow [4, 5]$
13. $[1] \rightarrow [2] \rightarrow [3] \rightarrow [4] \rightarrow [5]$

5 Number Division

Alice and Bob are playing a game with numbers. They have an array of integers $a_1, a_2, \ldots, a_n$ of length n and a special number k.

The game works like this:

1. Alice goes first and picks an index $1 \leq i \leq n$.
2. Then Bob picks a different index $1 \leq j \leq n, i \neq j$

- Alice wins if $|a_i - a_j|$ is not divisible by k.
- Otherwise, Bob Wins

Your task is to help Alice determine if she can win and, if so, which index i she can choose. Remember, Bob will play optimally to try to win.

The Absolute value of a number x is denoted by $|x|$ and is equal to $x \geq 0$, and $-x$ otherwise.

Input/Output

Input

Each test contains multiple test cases. The first line of input contains a single integer t ($1 \leq t \leq 10^3$) - the number of test cases.

- The first line of each test case contains two integers n and k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$) - the length of the array and the number k. It is guaranteed that the sum of n over all test cases does not exceed 10^5.
- The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$) - the elements of the array a.

Output

For each test case:

- Print -1 if Alice cannot win with optimal play.
- Otherwise, print the index i Alice should to guarantee a win. if multiple winning indices exist, print the **smallest** one.

Constraints

- For 20% of the tests, $t \leq 20$, $n \leq 20$, $k \leq 20$, $a_i \leq 20$.
- For 60% of the tests, $n \leq 100$, $k \leq 100$, $a_i \leq 100$.

Examples

Sample Input 1

```
7
3 2
1 2 3
2 2
15 16
4 2
1 2 4 5
2 2
15 15
5 3
10 7 4 5 3
5 3
1 31 15 36 55
1 3
5
```

Sample Output 1

```
2
1
-1
-1
4
-1
1
```

Explanation 1 In the first test case, if Alice chooses index 2 ($a_2 = 2$):

- If Bob picks 1, difference is $|2 - 1| = 1$ (not divisible by 2)
- If Bob picks 3, difference is $|2 - 3| = 1$ (not divisible by 2)

So Alice wins.

In the third test case, Alice can choose position 1 to win since $|15 - 16| = 1$ is not divisible by 2.

In the fourth test case, Alice cannot win. No matter which number she picks, Bob can always choose a number that makes their difference divisible by 2.

6 Wells

Wally and Wendy have invented a new game that they call well-racing.

Both Wally and Wendy start by climbing down a rope to the bottom of an H meter deep rectangular well, made of bricks of 1 meter in height. Their competition involves using different strategies to see if Wendy can reach the top of the well before she is caught by Wally.

Wally's strategy is to sit in a bucket attached to a rope, which his friend Weston uses to lift him out. Weston is able to lift Wally out of the well at a constant rate of 1 meter every second, so it always takes H seconds for Wally to get out of the well.

Wendy's climbing strategy involves jumping between a pair of the well's walls.

Wendy starts at the bottom of the left-hand wall and in each second, she can perform one of three possible movements.

1. Climb one meter up the wall she is currently on
2. Climb down one meter on the wall she is currently on (unless she is at the bottom of the well)
3. Jump across to the **opposite** wall, to a point which is **exactly** J meters higher than the point she jumped from

Wendy therefore leaves the well once she has climbed or jumped to a height of H meters *or higher*.

Wendy cannot climb or jump to every part of both walls - some of the bricks are covered in moss and too slippery to hold.

To avoid any accidental mid-air collisions, Wally and Wendy are take turns at moving for one second, with Wendy getting the first move.
I.e., the game starts with Wendy moving for 1 second while Wally stays still. In the next second, Wendy stays still while Wally is lifted for one second.

If Wally ends up at the same height as Wendy at any stage before Wendy escapes the well, Wally wins the game. If Wendy escapes without being caught, she wins the game.

For a given well, with a specific height and a certain set of slippery bricks, determine who the winner of the game will be if Wendy plays optimally.

Note that it may be impossible for Wendy to reach the top of the well under certain sets of slippery bricks - in this case the game is a forfeit.

Costraints

- $1 \leq H, J \leq 10\,000$

Input

Each test case will consist of T test cases, each being a separate well that Wally and Wendy will race in.

Each test case starts with the space separated integers H and J.

The H lines that follow describe the bricks on the left and right hand walls of the well. The character "." indicates a brick that is safe for Wendy to climb or jump to, and "#" indicates a slippery brick. Wendy starts the game holding onto the leftmost brick on the last line. Wally starts the game at the base of the well and moves to Wendy's starting height in his first move.

Output

For each of the T testcases, output one of the following strings:

- "WENDY" if Wendy will win the game when she plays optimally
- "WALLY" if Wally will win the game when Wendy plays optimally
- "FORFEIT" if it is impossible for Wendy to reach the top of the well

Examples

Sample Input 1

```
2
7 3
#.
.#
.#
#.
..
.#
..
6 2
..
##
.#
#.
..
.#
```

Sample Output 1

```
WENDY
WALLY
```