South African Computer Olympiad
Training Camp 3 2007
Day 2

Overview

Author	**Max Rabkin**	**Marco Gallotta**	**Keegan Carruthers-Smith and Bruce Merry**
Problem	**express**	**growth**	**stacking**
Source	express.c express.cpp	growth.c growth.cpp	stacking.c stacking.cpp
Input file	stdin	stdin	stdin
Output file	stdout	stdout	stdout
Time limit	1 second	2 seconds	1 second
Number of tests	10	10	10
Points per test	10	10	10
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Shortest Expressions

Author

Max Rabkin

Introduction

Fred the Manic Storekeeper has decided it is finally time for him to stop outsourcing his programming work to you, and learn some IT skills himself. He went looking for the cheapest compiler, but being ignorant of free software, he found a rather terrible one.

The compiler has a severe limit on the size of numeric literals; larger numbers must be built up from expressions. To make matters worse, it accepts mathematical expressions in postfix notation.

Postfix notation works like this:

- The expression is scanned from left to right.
- Each number encountered is pushed onto a stack.
- Each operator encountered pops its operands off the stack and pushes the result.

Fred still needs some help before he can write programs entirely on his own.

Task

There are some large numbers in Fred's program, which he has coded like this:

`five` ← `1 1 1 1 1 + + + +`

Unfortunately, the compiler is extremely slow, so long expressions make compilation times unbearable. Fred needs you to write a program that generates the shortest expression for a given number.

Example

The shortest expression (using numbers less than ten) that generates 19 is `1 2 9 * +`, or $1 + 2 * 9$ in infix notation.

Input (stdin)

The input consists of a single line containing two integers: N, the number for which you must generate an expression, and M, the maximum numeric literal allowed.

Sample input

```
19 9
```

Output (stdout)

The first line of the output consists of an integer L, the number of tokens (numbers and operators) in the expression.

The second line consists of L space-separated tokens. Each token is a non-negative integer, `+` or `*`.

Sample output

```
5
1 2 9 * +
```

Constraints

- $1 \leq N \leq 20000$
- $1 \leq M \leq 20000$

50% constraints

- $1 \leq N \leq 40$
- $1 \leq M \leq 40$

Time limit

1 second.

Scoring

If your output is incorrectly formatted, or the expression does not evaluate to N, or contains numbers greater than M you will score 0%.

If you produce the shortest possible expression, you will score 100%.

Otherwise, you will score $\frac{K}{L} \cdot 50\%$, where K is the shortest length.

Growth of the Shrubberies

Author

Marco Gallotta

Introduction

The Knights who say Ni have been very busy collecting shrubberies over the years. Recently they have been hearing rumours of large numbers disappearing from time to time. They suspect King Arthur is stealing them, but their collection has grown so large that it's not easy catching him out.

Before they can have him arrested, they need to prove his guilt. The first step in doing so is to determine the growth of their collection. If there is a negative growth, then they can arrest King Arthur for his crime. If they find a positive growth, then they can compare the growth rate to their incoming shrubberies and attempt to prove his guilt.

Task

Due to the large size of their shrubbery collection, they can only estimate its size at any one time. Comparing estimates directly is inaccurate. Fear not though, as one of the Knights has come up with a plan. To measure the growth, his plan is to look for subsets of fixed length from the data that indicate a strictly increasing number throughout the entire subset. A lack of such patterns should provide the perfect evidence against King Arthur.

The recorded history of growth goes back a long time and even will all the Knights working together they cannot find the patterns. It is therefore your task to determine the number of such patterns given the Knights' records.

The number of patterns found can get very large. The number found should therefore be given modulo 10^9, i.e. the remainder when divided by 10^9.

Example

Suppose the estimate of the number of shrubberies over the past four days were 20, 40, 10 and 50 and the Knights want to find strictly increasing subsets of length 2. There is an increasing growth from 20–40, 20–50, 40–50 and 10–50 giving a total of 4 subsets. 4 modulo 10^9 gives a final answer of 4.

Input (stdin)

The first line will contain two space-separated integers, N and K, the number of records and the subset length to search for. The next N lines each contain a single integer, V_i, which is the estimate on day i.

Sample input

```
4 2
20
40
10
50
```

Output (stdout)

Your output should consist of a single integer — the number of patterns found modulo 10^9.

Sample output

```
4
```

Constraints

- $1 \leq N \leq 10000$
- $1 \leq K \leq 100$
- $1 \leq V_i \leq 1000000$

50% constraints

- $1 \leq N \leq 1000$
- $1 \leq K \leq 10$

Time limit

2 seconds.

Scoring

A correct answer scores 100%, while an incorrect one scores 0%.

South African Computer Olympiad
Training Camp 3 2007
Day 2

Block Stacking

Author

Keegan Carruthers-Smith and Bruce Merry

Introduction

Mr Cutler is trying to convince Sir William of the Society of Putting Things on Top of Other Things is a pointless exercise. He is using the classic Stacking Blocks game as an example.

The game consists of an unlimited supply of blocks of C different colours. Over T seconds you can do one of the following every second:

- Put a block of some colour on top of the stack.
- Remove a block from the top of the stack. (There must be one)

Some colours are not aesthetically pleasing together, so they can never be on the stack at the same time. Colours are labelled 0 to $C-1$

The game starts with an empty stack. The game ends when the T seconds are up. The game must end with an empty stack.

Task

Count the number of possible ways this game can be played. Two schedules are different if at some point in time, the corresponding stacks have different sequences of colours.

Hopefully the large number of different possible combinations will convince Sir William that putting things on top of other things is pointless.

Example

If you have $C = 3$ colours, $T = 4$ seconds and colours 1 and 2 can not be on the stack at the same time, then the possible stack configurations are

- $t = 0$,
- $t = 1$ or 3, 0
- $t = 1$ or 3, 1
- $t = 1$ or 3, 2
- $t = 2$,
- $t = 2$, 0 0
- $t = 2$, 0 1
- $t = 2$, 0 2
- $t = 2$, 1 0
- $t = 2$, 1 1
- $t = 2$, 2 0
- $t = 2$, 2 2
- $t = 4$,

This gives a total of $K = 16$ schedules. Note that we do count the empty stack at $t = 0$ or at $t = 4$ even though they are implied to be empty. Also note that K gets modulo 1000001. $(10^6 + 1)$

Input (stdin)

Data is read in from standard input. The first line of input contains two space-separated integers, T and C. The second line contains a single integer N. The next N lines contain 2 space-separated integers, a_i and b_i. These represent colours that cannot be on the stack at the same time.

Sample input

```
4 3
1
1 2
```

Output (stdout)

You must write to standard output a single integer, $K\%1000001$, representing the number of possible ways to play the game.

Sample output

```
16
```

Constraints

- T is even
- $2 \leq T \leq 100$
- $1 \leq C \leq 10$
- $0 \leq N \leq \binom{C}{2} \leq 45$
- $0 \leq a_i, b_i \leq N - 1$

50% constraints

- $2 \leq T \leq 10$

Time limit

1 second.

Scoring

100% for the correct K, 0% otherwise.