MATHEMATICS DEPARTMENT

"A computer is the mathematicians best friend"

μ - Games
Mathematics Utrecht

December 2021

Rules:

The idea of this event is to gap the bridge between mathematics and programming. When working on these exercises, we hope the participant will get a better understanding of the underlying mathematical concepts used. You will not be required to do a lot of difficult programming. With array manipulation and basic functionality you should be able to solve all the exercises.

When working on these exercises, you must conform to the following rules.

- You are allowed to work in groups of maximum 4 persons.
- You will have three hours time.
- You can use no software packages.
- You can not look up any computer code that may help you solve the problem online.

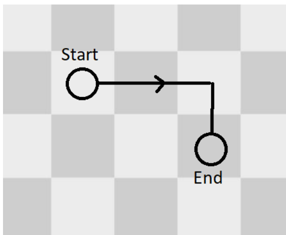
After the three hours, the solution to the exercises will be discussed and the solution has to be posted on the website *http://clover.science.uu.nl/dj*.

Problem 1: Super Knights

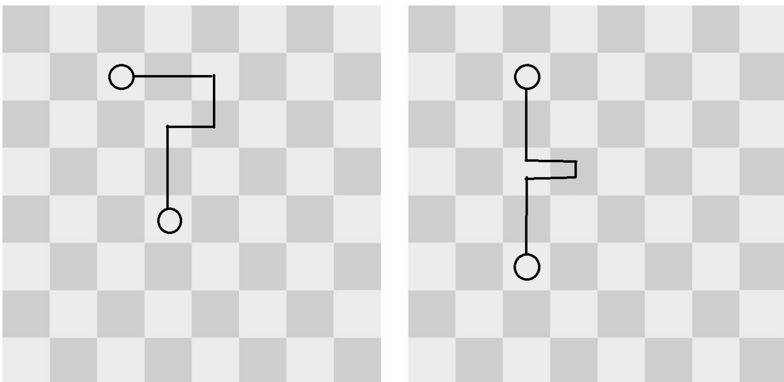
Difficulty: ★ ☆ ☆ ☆ ☆
Keywords: Logic

Exercise

Imagine a one-person game of chess in which your goal is to capture the king which is always at fixed position (x_k, y_k). To do this, you have access to N so-called Super Knights. As you maybe know, a move of a classic knight looks something like this:



Now a Super Knight combines two of these moves to make one move. So a Super Knights move looks something like this:



Now the problem is as follows: Given your N Super Knights are positioned on an $10^6 \times 10^6$ board at (x_i, y_i) for $1 \leq i \leq N$, which denotes the distance from the upper-left corner. How many can eventually capture the king after a finite number of moves?

Input

- A line with (x_k, y_k), the coordinate of the king,
- A line with the number N,
- N lines with the starting coordinates (x_i, y_i) of your starting knights, where the brackets, coordinates and comma are separated by spaces.

Output

- A line with the number of SuperKnights that are able to reach the king

Example 1

Input	Output
(2 , 1) 1 (2 , 6)	0

Example 2

Input	Output
(1 , 2) 2 (1 , 1) (3 , 4)	1

Problem 2: Lunch box

Difficulty: ★ ★ ☆ ☆ ☆
Keywords: Discrete optimization

Exercise

It is high time for a roadtrip! But you don't want to take too much with you, so you have one backpack to fill with the things you need. Now after you packed the essentials, you still have k units of space left for snacks. You have N different items of food, with each their own size r_i. But you notice something unusual about these sizes, namely:

$$r_i \geq \sum_{j=1}^{i-1} r_j,$$

for all $1 < i \leq N$. You prefer quantity above quality, so you want in total as much food as possible in terms of size. You now have to figure out which items to chose in order to archive this.

Input

- A line with the integer k,
- A line with the integer N,
- N lines with the integers r_1 up and until r_n

Output

- How many food you can fit into the backpack in terms of the size.

Example 1

Input	Output
6 3 1 3 6	6

Example 2

Input	Output
18 5 1 3 5 11 30	17

Problem 3: Lexicographic Permutations*

Difficulty: ★ ★ ☆ ☆ ☆
Keywords: Combinatorics

A permutation is an ordered arrangement of objects. For example, 3124 is one possible permutation of the digits $1, 2, 3$ and 4. If all of the permutations are listed numerically or alphabetically, we call it lexicographic order. The lexicographic permutations of 0, 1 and 2 are:

012 021 102 120 201 210

Given a set of distinct letters, $I \subseteq \{\text{a}, \text{b}, \ldots, \text{z}\}$, and a number N, give the N'th lexicographic permutation of I. You can assume $N < |I|!$

NB: The lexicographic ordering of the alphabet is given by the order they appear in there $(a < b, b < c, \ldots)$.

Input

- One line with integer N.
- One line with single space separated elements of set I.

Output

- One line containing a number given by the N'th lexicographic permutation of the letters in I with no spaces.

Example 1

Input	Output
3 a b c	bac

Example 2

Input	Output
50 c a e h b	cabhe

*Reference to be revealed later.

Problem 4: Unexpected Blowups

Difficulty: ★ ★ ★ ☆ ☆
Keywords: Analysis

Most of differential equations are not possible to solve analytically. As an alternative, one may attempt to construct an approximate solution numerically. There are many numerical methods that can be used to do so, each giving different answers for the same equation.

For example, consider the differential equation:

$$\begin{cases} \dfrac{dy(t)}{dt} = f(y(t)), \\ y(0) = y_0 \in \mathbb{C}. \end{cases} \tag{1}$$

Here f is a suitable function. Then the Euler method constructs numerical solution y_n at discrete time points $t_n = hn,\ n = 0, 1, \ldots$ by solving the recurrent equation

$$\frac{y_{n+1} - y_n}{h} = f(t_n, y_n), \text{ for } y_0 = y(0) \text{ and } h \in \mathbb{R}_{>0}\ .$$

Whereas according to the Trapezoidal method, approximate solution y_n is given by the solution of

$$\frac{y_{n+1} - y_n}{h} = \frac{1}{2} \cdot (f(t_n, y_n) + f(t_{n+1}, y_{n+1})), \text{ for } y_0 = y(0) \text{ and } h \in \mathbb{R}_{>0}.$$

Let us take $f(y) = ky$ with $k \in \mathbb{C}$ such that $\mathrm{Re}(k) < 0$. The differential equation (1) becomes:

$$\frac{dy(t)}{dt} = k \cdot y(t).$$

Note that our choice of k guarantees that $y(t)$ vanishes when $t \to \infty$. Interestingly, even if we know that $\lim_{t\to\infty} y(t) = 0$, this property in not necessary inherited in the numerical solution y_n. In fact, $\lim_{n\to\infty} y_n$ may not even exist. Determine for what k, h, the numerical methods agree with the exact solution of the differential equation asymptotically.

Input

- One line with the float h.
- One line with the real and the complex part of k. Bot parts are given by a float separated by a space.

Output

- One line with '1' if $\lim_{n\to\infty} y_n = 0$ where y_n is given by the Euler method, '0' otherwise.
- One line with '1' if $\lim_{n\to\infty} y_n = 0$ with y_n given by the Trapezoidal method, '0' otherwise.

Example

Input	Output
3	0
-2 5	1

Problem 5: Cryptic Message

Difficulty: ★ ★ ★ ★ ★
Key words: Number Theory

Exercise

Last μ-Games you made a new friend, but he left last year to Australia. To keep communicating you arranged "secret" protocol. Unfortunate for you, you lost some of your notes about the system. The thing you do remember, is how you and your friend decided to encode messages.

In order to encrypt a binary message of length n you will generate:

$$W = (w_1, ..., w_n)$$

a superincreasing sequence, which means that $w_k > \sum_{i=1}^{k-1} w_i$ for all $1 < k \leq n$. Also you chose some integer r, q.

$$q > \sum_{i=1}^{n} w_i$$

and

$$\gcd(r, q) = 1.$$

Together these form your private key (W, q, r).

With this you can create the public key $B = (b_1, ..., b_n)$, where $b_i = rw_i \mod q$ for all i, which you send to your friend across the world. Then your friend will encrypt his binary message $m = (m_1, ..., m_n)$ by calculating $c = \sum_{i=1}^{n} m_i b_i$.

Now how would you decode this message?

Hint: for this problem, you can use the following algorithm to find x, y such that for given a, b, $ax + by = \gcd(a, b)$:

```
function gcdExtended (a, b)
    if a == 0 :
        return b,0,1

    gcd,x1,y1 = gcdExtended(b%a, a)

    x = y1 - (b//a) * x1
    y = x1

    return gcd,x,y
```

Input

- A line with integer n,
- A line with your tuple $W = (w_1, .., w_n)$, given as n numbers $w_1 w_2 ... w_n$ separated by a space,
- A line with the number q,
- A line with the number r,
- A line with the coded message c you received, which is an integer

Output

- A binary string with the message your friend send you

Example 1

Input	Output
2 3 7 11 2 3	01

Example 2

Input	Output
4 1 3 5 11 22 7 31	1011

Problem 6: Zeros of Separable Functions.

Difficulty: ★ ★ ★ ★ ★
Keywords: Analysis, Logic, Discrete Mathematics

A multivariate function $f(x_1, x_2, \dots, x_n)$ is called separable when it can be represented as a product of lower-dimensional functions. Consider a multivariate function that is separable into k linear terms of the form:

$$f(x_1, x_2, \dots, x_n) = \prod_{j=1}^{k} (\alpha_{j,1} x_{i_{j,1}} + \alpha_{j,2} x_{i_{j,2}} - \min(\alpha_{j,1}, 0) - \min(\alpha_{j,2}, 0)), \; (x_1, x_2, \dots, x_n) \in [0,1]^n$$

with $\alpha_{j,1}, \alpha_{j,2} \in \{-1, 1\}$ and $i_{j,1}, i_{j,2} \in 1, \dots, n$ for $j = 1, \dots, k$.

It is easy to see that f is always positive on the interior of the domain, in $(0,1)^n$. Your task is to determine whether specified f vanishes at *all* corners of the domain hypercube or not, that is whether

$$f(x_1, \dots, x_n) = 0 \text{ for all } (x_1, \dots, x_n) \in \{0,1\}^n.$$

Input

- A line with an integer $1 \leq n \leq 100$.
- A line with an integer $1 \leq k \leq 10000$.
- k lines containing entries $(\alpha_{j,1}, \alpha_{j,2}, i_{j,1}, i_{j,2})$.

Output

- 0 – if f is non-vanishing at least at one vertex of the domain cube.
- 1 – if f vanishes at all vertices of the domain cube.

Example 1

Input	Output
2 2 1 1 1 2 1 -1 1 2	0

Example 2

Input	Output
3 4 1 1 1 2 1 -1 1 2 -1 1 1 3 -1 -1 1 3	1