Programming Assignments

Competitive Programming: Core Skills

September 28, 2019

Welcome to the "Competitive Programming: Core Skills" MOOC at Coursera! This document contains the statements of all programming assignments of the course. There are six programming assignments (one for each week), each containing four problems. To pass a programming assignment, you need to solve at least two problems from it. For the first two problems in each assignment, we provide starter solutions in `C++`, `Java`, and `Python`.

When you pass a programming assignment (i.e., when you successfully solve any two of its problems), you will get access to a locked reading that contains solution to the fourth problem of this programming assignment. We strongly encourage you to take a look at these solutions *only after* you've tried to solve this problem yourself: it will be more useful for you if you solve the problem without reading its solution (not to mention satisfaction and self-confidence that you get upon solving :)). If you decide to read the solution, make sure to implement it and to submit to the grading system. This will allow you to understand it even better, and to improve your implementation skills.

You may submit solutions in any of the ten supported programming languages listed in the end of this document. We believe that the time/memory limits allow to solve each of the problems in all of these languages, but we have only checked this for `C++`, `Java`, and `Python` (hence, we provide no guarantees for other languages, use them at your own risk).

The time limit "n seconds" in a problem statement means the following: n seconds for `C`, `C++`, `Rust`; $1.5n$ seconds for `Java` and `C#`; $3n$ seconds for `Scala`; $5n$ seconds for `JavaScript`, `Python`, `Ruby`.

Contents

1 Week 1: Programming Competitions

1.1 Addition and Subtraction

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given two integers x and y, construct an infinite sequence of integers $A = \{a_0, a_1, a_2, \dots\}$ as follows: $a_0 = 0$ and for every $i \geq 1$, $a_{2i-1} = a_{2i-2} + x$ and $a_{2i} = a_{2i-1} - y$.

Given three integers x, y and z, find the index of the first occurrence of z in A or report that z does not appear in A.

For example, if $x = 2$, $y = 1$ and $z = 3$, then $A = (0, 2, 1, 3, 2, 4, 3, \dots)$ and the answer is 3 ($a_3 = 3$ and this is the first occurrence of 3 in A). If $x = 2$, $y = 0$ and $z = 3$, then $A = (0, 2, 2, 4, 4, 6, 6, \dots)$ and the answer is -1 (there is no occurrence of 3 in A).

Input

Three integers x, y and z ($0 \leq x, y, z \leq 1000$) separated by spaces.

Output

The first position of z in A or -1, if there is no occurrence of z in A.

Examples

standard input	standard output
2 1 3	3
2 0 3	-1

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/1/threads/d9f6Tom8Eei5AwpbAz2WpA`

1.2 Erasing Maximum

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Let $A[1..n]$ be an array of integers. Output the same array without its maximum element. If there are several maximum elements, get rid of the third. It is guaranteed that the input array A has either a unique maximum element or at least three maximum elements.

Input

The first line contains an integer n ($2 \le n \le 100$), the length of the array. The second line contains integers $A[1], A[2], \ldots, A[n]$ ($1 \le A[i] \le 100$, $1 \le i \le n$).

Output

Output $n - 1$ integers separated by spaces.

Examples

standard input	standard output
3 1 3 2	1 2
7 4 1 4 2 4 3 4	4 1 4 2 3 4

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/1/threads/B9xDNYm9EeigjhIs5jsscA`

1.3 Increment

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given a large non-negative integer x, find the number of decimal digits in $x + 1$.

Input

A non-negative integer x ($0 \le x \le 10^{1\,000\,000}$) with no leading zeroes.

Output

The number of decimal digits in $x + 1$.

Examples

standard input	standard output
1	1
9	2

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/1/threads/ICQVzIm9EeiwzxKiAWTjZA`

1.4 Straight Flush

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

The deck of 52 French playing cards is the most common deck of playing cards used today. It includes thirteen ranks of each of the four French suits: clubs (`C`), diamonds (`D`), hearts (`H`), and spades (`S`). Each suit includes an ace (`A`), a king (`K`), a queen (`Q`), and a jack (`J`), each depicted with a symbol of its suit; and ranks two (`2`) through ten (`T`), with each card depicting that many symbols (pips) of its suit.

A **straight flush** is a poker hand containing five cards of sequential rank, all of the same suit, such as `QH JH TH 9H 8H` (a "queen-high straight flush"). As part of a straight flush, an ace (`A`) can rank either above a king or below a two. So an ace can rank either high (e.g., `AH KH QH JH TH` is an ace-high straight flush) or low (e.g., `5D 4D 3D 2D AD` is a five-high straight flush), but cannot rank both high and low in the same hand (e.g. `QS KS AS 2S 3S` is not a straight flush). Thus, there are 40 possible straight flush hands when using a standard 52-card deck.

Given 5 different cards, check whether they constitute a straight flush.

Input

Five different cards separated by spaces. Each card is specified by two symbols: the first one is a rank and the second is a suit. Suites are marked with letters `C`, `D`, `H`, and `S`. Ranks are marked with symbols `2`, `3`, ..., `9`, `T`, `J`, `Q`, `K`, and `A`.

Output

Output `YES`, if the given cards form a straight flush, otherwise output `NO`.

Examples

standard input	standard output
`2D 5D 3D 4D 6D`	`YES`
`AD KH QH JS TC`	`NO`

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/1/threads/On3-hIm9EeiukwpCJ4eUBA`

2 Week 2: Correctness First

2.1 The Cheapest Permutation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Let A be a symmetric $n \times n$ matrix of integers (symmetric means that $A[i][j] = A[j][i]$ for all $1 \le i, j \le n$). For a permutation π of integers from 1 to n, define its *cost* as

$$\sum_{i=1}^{n-1} A[\pi_i][\pi_{i+1}]$$

(hence, $A[i][j]$ can be viewed as the cost of placing integers i and j in adjacent positions). Your goal is to find a permutation of minimum cost.

Input

The first line contains an integer n ($1 \le n \le 9$). Each of the next n lines contains n integers $A[i][j]$ ($0 \le A[i][j] \le 1\,000$). It is guaranteed that $A[i][i] = 0$ for all $1 \le i \le n$ and $A[i][j] = A[j][i]$ for all $1 \le i, j \le n$.

Output

Output a permutation of minimum cost (integers from 1 to n in some order separated by spaces). In case of multiple correct answers output any of them.

Examples

standard input	standard output
3 0 1 2 1 0 4 2 4 0	2 1 3
4 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0	1 2 3 4

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/2/threads/5D4eLIprEeigsgojVVERSg`

2.2 The King

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

You goal is to solve the chess king puzzle. Recall that a chess king moves one square in any direction (horizontally, vertically, or diagonally). Your goal is to place as many king as possible on an $r \times c$ board subject to the following two conditions:

- There is at most one king in any cell of the board.
- Each king has at least one possible move into a free cell on the board.

Input

Two integers r and c ($1 \leq r, c \leq 100$) separated by spaces.

Output

The maximum number of kings.

Examples

standard input	standard output
1 2	1
3 3	8

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/2/threads/Qa3LuIpsEei_ZhJROJlweg`

2.3 Sum of Minimums

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Let $A[1..n]$ be an array of integers. For $1 \leq l \leq r \leq n$, let

$$m(l, r) = \min\{A[l], A[l+1], \ldots, A[r]\}$$

be the minimum value of A on an interval $[l, r]$. E.g., for $A = [3, 1, 2, 3]$, $m(1, 1) = 3$, $m(2, 2) = 1$, $m(3, 3) = 2$, $m(4, 4) = 3$, $m(1, 2) = 1$, $m(2, 3) = 1$, $m(3, 4) = 2$, $m(1, 3) = 1$, $m(2, 4) = 1$, and $m(1, 4) = 1$.

Given an array A, find the sum of minimums for all intervals (i.e., $\sum_{1 \leq l \leq r \leq n} m(l, r)$).

Input

The first line contains an integer n ($1 \leq n \leq 2\,000$), the length of the array. The second line contains integers $A[1], A[2], \ldots, A[n]$ ($1 \leq A[i] \leq 1\,000\,000$, for all $1 \leq i \leq n$).

Output

The sum of minimums.

Example

standard input	standard output
4 3 1 2 3	16

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/2/threads/X2RNCYpsEeiQAxLz_acxuA`

2.4 Expression Evaluation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Evaluate an arithmetic expression with only two types of arithmetic operations: addition and subtraction.

Input

The expression s to be evaluated, a non-empty string containing only digits and symbols of arithmetic operations + (plus) and - (minus). The length of s does not exceed 50 000.

It is guaranteed that the first symbol of s is a digit and each number in the expression is a non-negative integer with no more than six decimal digits.

Output

The evaluation result.

Examples

standard input	standard output
1+2+3+4	10
1-2+3-4	-2

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/2/threads/fjtZ7IpsEeixcg7-I5Ek7A`

3 Week 3: Common Struggles

3.1 Compare Sums

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given two arrays of floats A and B, check which one has a larger sum of elements.

Input

The first line contains an integer n ($1 \le n \le 100$), the length of the arrays.

The second line contains floating point numbers $A[1], A[2], \ldots, A[n]$ ($0.001 \le A[i] \le 1\,000$, for all $1 \le i \le n$), the elements of A.

The third line contains floating point numbers $B[1], B[2], \ldots, B[n]$ ($0.001 \le B[i] \le 1\,000$, for all $1 \le i \le n$), the elements of B.

Each floating point number has exactly 3 digits after the decimal point.

Output

Output `SUM(A)=SUM(B)`, if both sequences have equal sum of elements. Output `SUM(A)>SUM(B)`, if sequence A has larger sum. Otherwise output `SUM(A)<SUM(B)`. Avoid any white spaces.

Examples

standard input	standard output
2 1.500 1.500 1.000 2.000	SUM(A)=SUM(B)
1 2.000 1.123	SUM(A)>SUM(B)
3 1.000 2.000 3.000 1.001 2.001 3.001	SUM(A)<SUM(B)

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/3/threads/9QucVIpsEei-VApvVhRkHg`

3.2 Round Up

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given a rational number $\frac{x}{y}$, find the minimum integer z such that $z \geq \frac{x}{y}$.

Input

Two integers x and y ($-10^9 \leq x, y \leq 10^9$, $y \neq 0$).

Output

The required value z.

Examples

standard input	standard output
1 2	1
1 -2	0
10 10	1

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/3/threads/kGvD1optEeigsgojVVERSg`

3.3 Yet Another Sum

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given a sequence of integers $x_1, x_2, \ldots, x_n$, compute the sum

$$s = \sum_{i=1}^{n} \left(x_i + \frac{1}{x_i} \right) .$$

Pay attention to the precision of the resulting *double*.

Input

The first line contains an integer n ($1 \le n \le 50$), the length of the sequence.

The second line contains integers $x_1, x_2, \ldots, x_n$ ($-10^9 \le x_i \le 10^9$, $x_i \neq 0$, for all $1 \le i \le n$).

Output

A floating-point number s. The answer will be graded as correct, if absolute or relative error does not exceed 10^{-9}.

Examples

standard input	standard output
1 3	3.3333333333
1 -4	-4.2500000000
4 -2 -3 1 2	-1.3333333333

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/3/threads/ruRdToptEeigsgojVVERSg`

3.4 Binary Knapsack

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

The *knapsack problem* is a classical problem in combinatorial optimization: Given a set of n items, each with a weight w_i and a value v_i, determine the items to include in a collection so that the total weight is less than or equal to a given limit w and the total value is as large as possible.

Your goal is to solve a special case of the knapsack problem with additional constraint: the weight of each item is a power of 2.

Find the the set of items with the maximum total value so that their total weight is at most w.

Input

The first line contains two integers n and w ($1 \le n \le 100$, $0 \le w \le 2^{30}$), the number of items and the maximal total weight of items in the knapsack.

Each of the next n lines contains two integers w_i and v_i ($1 \le w_i \le 2^{30}$, $0 \le v_i \le 10^6$), the weight and the value of i-th item in the set. Each w_i is a power of 2.

Output

The maximum total value of items in the selected set.

Examples

standard input	standard output
3 5 1 3 1 2 1 1	6
5 10 1 4 2 5 1 2 4 6 8 12	18
1 5 8 3	0

Note

Each item can be selected at most once.

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/3/threads/1-6mIoptEeigsgojVVERSg`

4 Week 4: Common Struggles 2

4.1 The Most Frequent Symbol

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

For any string S you can easily find the most frequent symbol. What about queries for some substring of S? Can you find the most frequent symbol quickly?

You are given a string $S = s_1 s_2 \dots s_{|S|}$ and a set of queries, where each query is a pair of indices (l, r), $1 \le l \le r \le |S|$. For each query (l, r), find the most frequent symbol in $s_l s_{l+1} \dots s_r$.

Input

The first line contains a strings S ($1 \le |S| \le 50\,000$). The string S consists of small Latin letters only.

The second line contains an integer Q ($1 \le Q \le 50\,000$), the number of queries.

Each of the next Q lines contains two integers l and r ($1 \le l \le r \le |S|$), positions of the first and the last symbols in the substring.

Output

For each query, print a single line with the most frequent symbol in the substring. In case of multiple most frequent symbols output any of them.

Examples

standard input	standard output
abacaba 3 1 1 1 7 2 4	a a c
abba 6 1 1 2 2 1 2 2 3 1 1 2 4	a b b b a b

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/4/threads/tEnUJJDzEeiEoRLp7BO6cA`

4.2 Maximal Distance

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Find a pair of points on the line with maximum distance between them:

$$\text{Distance}(a, b) = |x_a - x_b|.$$

To make this problem more interesting, let's find such a pair after each step of adding points to the line.

Input

The first line contains an integer n ($1 \le n \le 100\,000$), the number of points on the line.

The i-th of the next n lines contains coordinates x_i of the i-th point ($0 \le x_i \le 10^9$ is integer).

All points are different.

Output

Print n lines. The i-th line should contain two numbers f_i and s_i ($1 \le f_i, s_i \le i$), the indices of two points with maximum distance among the first i points.

In case of multiple correct pairs of points, print any of them.

Examples

standard input	standard output
3 1 2 3	1 1 1 2 1 3
5 3 2 1 50 49	1 1 1 2 1 3 3 4 3 4

4.2.1 Manhattan Distance

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Find a pair of points on the Cartesian plane with maximum Manhattan distance between them:

$$\text{ManhattanDistance}(a, b) = |a_x - b_x| + |a_y - b_y|.$$

To make this problem more interesting, let's find such a pair after each step of adding points to the plane.

Input

The first line contains an integer n ($1 \le n \le 100\,000$), the number of points on the plane.

The i-th of the next n lines contains coordinates x_i, y_i of the i-th point ($0 \le x_i, y_i \le 10^9$ are integers). All points are different.

Output

Print n lines. The i-th line should contain two numbers f_i and s_i ($1 \le f_i, s_i \le i$), the indices of two points with maximum Manhattan distance among the first i points.

In case of multiple correct pairs of points, print any of them.

Examples

standard input	standard output
3 1 1 2 1 1 3	1 1 2 1 2 3
5 2 2 1 3 1 1 3 1 3 3	1 1 1 2 3 2 4 2 4 2

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/4/threads/3f617pDzEei2sgp-AWth6A`

4.3 Multiset

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Consider a multiset of integers S, the union of n closed intervals of positive integers:

$$S = [l_1..r_1] \cup [l_2..r_2] \cup \cdots \cup [l_n..r_n]$$

(recall that a closed interval $[l..r]$ contains integers $\{l, l+1, \ldots, r\}$).

Let D be the set of unique integers in S. For each x in D, find the number of occurrences of x in S.

Input

The first line contains an integer n ($1 \le n \le 100\,000$), the number of intervals in the union.

Each of the next n lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le 100\,000$), the left and right boundaries of the i-th interval.

Output

For each integer x in D, print two integers on a separate line: x and its number of occurrences in S.

Examples

standard input	standard output
3 1 1 3 4 6 9	1 1 3 1 4 1 6 1 7 1 8 1 9 1
5 1 10 2 9 3 8 4 7 5 5	1 1 2 2 3 3 4 4 5 5 6 4 7 4 8 3 9 2 10 1

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/4/threads/DnoN4pD0Eei65w7hocEa4A`

4.4 Maximal Sum Subarray

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Let $A[1..n]$ be an array of integers. For all i from 1 to n find a subarray with maximum sum that covers the position i (more formally, for every i, find the largest value $A[l] + A[l+1] + \cdots + A[r]$ among all pairs of indices l and r such that $1 \le l \le i \le r \le n$).

Input

The first line contains an integer n ($1 \le n \le 100\,000$), the number of elements in A.

The second line contains integers $A[1], A[2], \ldots, A[n]$ ($-10^6 \le A[i] \le 10^6$).

Output

Print n integers separated by spaces. The i-th of them should be equal to the maximal sum of subarray among all that cover the position i in A.

Examples

standard input	standard output
3 -1000000 1000000 -1000000	0 1000000 0
4 1 2 3 4	10 10 10 10
5 2 -3 -3 -3 2	2 -1 -3 -1 2

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/4/threads/OSqNmpD0EeibUg4cVkqK9g`

5 Week 5: Dynamic Programming

5.1 Longest Increasing Subsequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Let $A[1..n]$ be an array of integers. Find the largest k ($1 \le k \le n$) such that there is a sequence of indices $i_1, i_2, \ldots, i_k$ ($1 \le i_1 < i_2 < \cdots < i_k \le n$) satisfying $A[i_1] < A[i_2] < \cdots < A[i_k]$.

For example, an array $A = [5, 3, 2, 4, 6, 1]$ has two longest increasing subsequences $(3, 4, 6)$ and $(2, 4, 6)$.

Input

The first line contains an integer n ($1 \le n \le 2\,000$), the number of elements in A.

The second line contains integers $A[1], A[2], \ldots, A[n]$ ($1 \le A[i] \le 1\,000\,000$).

Output

The length of the longest increasing subsequence.

Examples

standard input	standard output
6 5 3 2 4 6 1	3
6 1 1 2 2 3 3	3
5 5 4 3 2 1	1

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/5/threads/duYUOpDOEei2sgp-AWth6A`

5.2 Edit Distance

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given two strings u and w.

Find the weighted edit distance between u and w, where the cost of inserting is I, the cost of deletion is D, and the cost of substitution is S.

Input

The first line contains two integers n and m ($1 \le n, m \le 1\,000$), the length of strings u and w, respectively.

The strings u and w are given in the second and third lines, respectively. Both strings consist of small Latin letters only.

The forth line contains three integers I, D and S ($1 \le I, D, S \le 100$).

Output

The minimal cost to transform u to w by single symbol insertions, deletions, and substitutions.

Examples

standard input	standard output
7 8 editing distance 1 1 1	5
7 8 editing distance 1 1 100	7
7 8 editing distance 100 1 1	105
7 8 editing distance 1 100 1	6

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):

https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/5/threads/qQjcAJD0EeiZbRJpg5oTDA

5.3 Sum of Digits

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given two integers S and L.

Consider all *non-negative* decimal integers without leading zeros of length L. Find how many of them has sum of digits equal to S.

Input

Two integers S and L ($0 \le S \le 162$, $1 \le L \le 18$).

Output

The number of integers with L decimal digits with sum of digits equal to S.

Examples

standard input	standard output
5 2	5
1 1	1
10 10	48619

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/5/threads/yM5AaZD0Eei2sgp-AWth6A`

5.4 Make It Sorted

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Let $A[1..n]$ be an array of integers. In one step, you are allowed to add either 1 or -1 to any element of the array. What is the minimum number of steps required to make the array sorted in non-decreasing order ($A[1] \leq A[2] \leq \cdots \leq A[n]$)?

Input

The first input line contains one integer n ($1 \leq n \leq 2\,000$), the number of elements in A.

The second input line contains integers $A[1], A[2], \ldots, A[n]$ ($1 \leq A[i] \leq 1\,000$).

Output

The minimum number of steps required to make the sequence sorted.

Examples

standard input	standard output
3 1 2 1	1
2 5 10	0
7 1 4 2 3 1 4 4	4

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/5/threads/_d6J8ZD0Eei65w7hocEa4A`

6 Week 6: Dynamic Programming 2

6.1 Knapsack

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Recall the knapsack problem: Given a set of n items, each with a weight w_i and a value v_i, determine the items to include in a collection so that the total weight is less than or equal to a given limit W and the total value is as large as possible. Each item can be taken at most once (i.e., there is a single copy of every item).

Input

The first line contains two integers n and W ($1 \le n \le 100$, $0 \le W \le 1\,000$), the number of items and maximal total weight of items in a knapsack.

Each of the next n lines contains two integers w_i and v_i ($1 \le w_i \le 1\,000$, $0 \le v_i \le 1\,000$), the weight and the value of the i-th item.

Output

The first line should contain an integer k ($0 \le k \le n$), the number of items in a collection.

The second line should contain k different integers from 1 to n, the indices of included items.

In case of several optimal solutions, output any of them.

Examples

standard input	standard output
2 2 2 2 1 4	1 2
5 5 1 1 2 3 3 2 5 4 4 5	2 1 5
3 100 52 99 51 50 49 50	2 2 3

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/6/threads/Kq_GcZD1EeiX3AozPw-ZEg`

6.2 Chain Matrix Multiplication

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given a sequence of n matrices $A_0, \ldots, A_{n-1}$ to be multiplied. The size of the matrix A_i is $m_i \times m_{i+1}$.

Find an order of multiplication minimizing the total cost of multiplication. Assume that the cost of multiplying $p \times q$ and $q \times r$ matrices is equal to $p \cdot q \cdot r$ (this is the number of integer multiplications of a straightforward matrix multiplication algorithm).

Input

The first line contains one integer n ($2 \le n \le 150$), the number of matrices to be multiplied.

The second line contains $(n + 1)$ integers $m_0, m_1, \ldots, m_n$ ($1 \le m_i \le 1\,000$), the size of the i-th matrix is $m_i \times m_{i+1}$.

Output

The minimum cost of multiplying the given n matrices.

Examples

standard input	standard output
2 10 20 10	2000
3 50 20 1 10	1500
10 1 2 3 4 5 6 5 4 3 2 1	138

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/6/threads/RjDLAJD1EeiEoRLp7BO6cA`

6.3 Longest Common Subsequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given two sequences $A = (a_0, \ldots, a_{n-1})$ and $B = (b_0, \ldots, b_{n-1})$ of the same length.

Find the largest integer k such that there exist two sequences of indices $0 \le i_1 < i_2 < \cdots < i_k < n$ and $0 \le j_1 < j_2 < \cdots < j_k < n$ satisfying $a_{i_1} = b_{j_1}, a_{i_2} = b_{j_2}, \ldots, a_{i_k} = b_{j_k}$.

Input

The first line contains an integer n ($1 \le n \le 1\,000$), the length of A and B. The second line contains n integers a_i ($1 \le a_i \le 1\,000$), the sequence A. The third line contains n integers b_j ($1 \le b_j \le 1\,000$), the sequence B.

Output

The first line should contain k, the length of the longest common subsequence. The second line should contain indices $i_1, i_2, \ldots, i_k$. The third line should contain indices $j_1, j_2, \ldots, j_k$.

Examples

standard input	standard output
2 1 2 1 2	2 0 1 0 1
5 1 2 3 4 5 1 3 2 4 4	3 0 1 3 0 2 4
6 1 2 3 3 4 6 1 6 3 3 2 4	4 0 2 3 4 0 2 3 5

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/6/threads/aYSs2JD1EeiUaw4TMd55tA`

6.4 Maximal Sum Square

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given an $n \times n$ matrix A of integers and an integer k.

Find a $k \times k$ submatrix of A with maximum sum of elements.

Input

The first line contains two integers n and k ($1 \leq k \leq n \leq 700$), dimensions of the input matrix and submatrix, respectively.

The i-th of next n lines contains n integers a_{ij} ($0 \leq a_{ij} \leq 1\,000$), the elements of the i-th row of A.

Output

The maximum sum of elements of a $k \times k$ submatrix.

Examples

standard input	standard output
3 2 10 2 10 1 1 1 10 1 13	16
2 1 4 1 1 2	4
5 3 1 2 3 4 5 5 4 3 2 1 1 1 1 1 1 2 1 2 1 2 6 3 4 1 5	21

Please discuss this problem at the following forum thread (make sure to review the forum rules at the end of the document before posting):
`https://www.coursera.org/learn/competitive-programming-core-skills/discussions/weeks/6/threads/hODDnZD1EeioiQpfwnpH6g`

7 Appendix

7.1 Compiler Flags

The following compilers and flags are used while grading your solutions.

C (`gcc 5.2.1`). File extensions: `.c`. Flags:
`gcc -pipe -O2 -std=c11 <filename> -lm`

C++ (`g++ 5.2.1`). File extensions: `.cc`, `.cpp`. Flags:
`g++ -pipe -O2 -std=c++14 <filename> -lm`

C# (`mono 3.2.8`). File extensions: `.cs`. Flags:
`mcs`

Haskell (`ghc 7.8.4`). File extensions: `.hs`. Flags:
`ghc -O2`

Java (`Open JDK 8`). File extensions: `.java`. Flags:
`javac -encoding UTF-8`
`java -Xmx1024m`

JavaScript (`Node v6.3.0`). File extensions: `.js`. Flags:
`nodejs`

Python 2 (`CPython 2.7`). File extensions: `.py2` or `.py` (a file ending in `.py` needs to have a first line which is a comment containing "python2"). No flags:
`python2`

Python 3 (`CPython 3.4`). File extensions: `.py3` or `.py` (a file ending in `.py` needs to have a first line which is a comment containing "python3"). No flags:
`python3`

Ruby (`Ruby 2.1.5`). File extensions: `.rb`. No flags:
`ruby`

Rust (`Rust 1.7`). File extensions: `rs`. No flags:
`rust`

Scala (`Scala 2.11.6`). File extensions: `.scala`. No flags:
`scalac`

7.2 Forum Rules

- Follow etiquette rules: respect other learners, make your post valuable for others (in particular, before asking check whether someone else has already asked it; keep your post as short as possible). See more common sense rules to follow: `http://blogs.onlineeducation.touro.edu/15-rules-netiquette-online-discussion-boards/`
- Please do not post solutions of programming challenges or their algorithmic parts (this violates the Coursera honor code), even if they do not pass the tests.

- We encourage you to post starter files for various programming languages. A starter file should only read the input data and write the output data.
- If your first attempt to solve a problem failed, and then you debugged and fixed your program, you may want to share this bug. Examples: "Remember to use `//` for integer division for `Python3`", "Remember that a class name and a file name should match for Java", "Remember to use `long long` type if you are dealing with large numbers for `C++`", but please do not post the code, just mention the "idea" of the bug.
- You may want also to help other learners by posting small tests that can help to catch a bug. In this case, please comment how did you come up with the test, so that others can learn from your creativity.
- Please note that instructors are not going to reply to the following typical questions:
  - "My program runs fine on my local machine, but fails in the grader. Could you fix the grader?"

    All the learners' computers are slightly different, so the only way to make grading fully objective is to grade all solutions on the same server. Sometimes, the result of compiling and running the same program is different on different computers because of different compilers, different compiler settings, different operating system or just some bugs in the program code that lead to undefined behavior. However, it is always possible to write a correct program that works correctly both on your local machine and in our grader. You will have to write such a program to pass. We do our best to specify all the important parameters of the grader that we know of here.
  - "I use exactly the same compiler and flags as in the grader, but my program has different outputs on my machine and in the grader. Could you check the grader?"

    This usually happens with buggy programs that run into undefined behavior.
  - "Any hints about test 42?"

    Recall that we hide the test cases intentionally. Please test and stress test you program to fix it. If you have already tested a lot (considered all corner cases that you can imagine, constructed a set of manual test cases, applied stress testing), but your program still fails and you are stuck, try to ask for help on the forum. We encourage you to do this by first explaining what kind of corner cases you have already considered (it may happen that when writing such a post you will realize that you missed some corner cases!) and only then asking other learners to give you more ideas for tests cases.
  - "How to compile/run a starter file?"

    Please note that though this class has many programming exercises, it is not a programming class. We expect you to know how to program in a programming language of your choice.