Problem Set

Please check that you have 12 problems that are spanned across 21 pages in total (including this cover page).

A. All You Need is Dating	(2 pages)	
B. Balanced String	(1 page)	Korean translation available
C. Byte Coin	(2 pages)	Korean translation available
D. Canal	(2 pages)	
E. Choreography	(3 pages)	
F. Dryer	(2 pages)	
G. Enumeration	(1 page)	
H. Four Squares	(1 page)	Korean translation available
I. Registration	(1 page)	Korean translation available
J. Star Trek	(2 pages)	
K. Steel Slicing	(2 pages)	
L. Two Machines	(1 page)	Korean translation available

The memory limits for the twelve problems are all the same, 512MB.

Problem A
All You Need is Dating
Time Limit: 1 Second

Anne likes to arrange a blind date between people and opens up a startup company called 'Anne Dating Center'. One day, she receives an order from IC school who wants to set up a blind dating festival with the PC school students. Anne first collects the information of all PC students who want to participate the dating festival. Then she distributes the collected information to the IC students and asks IC students to hand in their preferred date partners from PC school. An IC student can have several preferred partners from PC school. One unexpected request from both IC and PC students is the numbers of the minimum and maximum dates. For the sake of the dating festival, each student has a minimum expectation on the number of dates. On the other hand, due to an upcoming exam, a student has a limit on the maximum number of dates. In summary, all the students from both IC school and PC school have their own minimum and maximum numbers of dates that they would participate in. In addition, a student may have several dates during the festival, yet a pair of an IC student and a PC student can have at most one date between two students.

For the blind dating between m IC students and n PC students, the following three restrictions are given. (1) k preferred partner lists from IC school students, (2) minimum and maximum numbers of dates that IC students want, and (3) minimum and maximum numbers of dates that PC students want. Note that Anne can set up a date between an IC student u and a PC student v only when v is in the preferred partner list of u. Therefore, if there is an IC student u whose number of preferred partners is smaller than the minimum number of dates, then Anne cannot arrange a date for u. In summary, given the information, Anne needs to arrange the date between IC and PC students such that each date should be set up only when the pair is in the preferred partner lists, and the number of scheduled dates for each student should be within his/her own minimum and maximum expectations. Write a program to find the maximum number of dates that Anne can arrange such that all three restrictions are satisfied.

Input
Your program is to read from standard input. The input starts with a line containing three integers, m, n and k ($1 \le m \le 100$, $1 \le n \le 100$, $1 \le k \le 10{,}000$), where m is the number of IC school students, n is the number of PC school students, and k is the number of preferred partner lists. Each student from IC school is given a unique number from 1 to m and each student from PC school is given a unique number from 1 to n. These unique numbers are student IDs. The second line contains m nonnegative integers, each of which is the i-th IC student's minimum number of expected dates. The third line contains m nonnegative integers, each of which is the i-th IC student's maximum number of expected dates. Similarly, the fourth line contains n nonnegative integers for the PC student's minimum expected dates, and the fifth line contains n nonnegative integers for the PC student's maximum expected dates. The minimum and maximum numbers of dates for each student are at least 0, and at most n for IC students and m for PC students. In the following k lines, each line contains two student IDs such that the first is an IC student's ID and the second is the IC student's preferred PC student ID.

Output
Your program is to write to standard output. Print exactly one line. The line should contain the maximum number of dates that Anne can arrange. If Anne cannot arrange a dating schedule that satisfies all three restrictions, print -1.

The following shows sample input and output for three test cases.

Sample Input 1

```
3 3 3
0 0 0
3 3 3
0 0 0
3 0 3
1 2
2 2
3 2
```

Output for the Sample Input 1

```
0
```

Sample Input 2

```
5 4 5
1 1 1 1 1
1 1 1 1 1
1 1 1 1
1 1 2 1
1 1
2 2
3 3
1 3
5 4
```

Output for the Sample Input 2

```
-1
```

Sample Input 3

```
6 5 11
1 1 1 1 1 1
2 5 1 2 5 2
1 1 1 1 1
3 2 3 2 3
1 3
2 5
4 1
3 4
5 2
6 4
1 2
4 2
1 4
3 3
6 5
```

Output for the Sample Input 3

```
9
```

Problem B
Balanced String
Time Limit: 0.5 Seconds

The difference between the number of $0's$ and the number of $1's$ of a binary string 0101101 is less than or equal to 1. Furthermore, all substrings of 0101101 including the first character, i.e. 0, 01, 010, 0101, 01011, 010110, and 0101101 have the difference less than or equal to 1.

A binary string S is considered a **balanced string** if the difference of the number of 0 and the number of 1 is less than or equal to 1 for all substrings of S that include the first character. Note that S itself is a substring of S.

Given a positive integer n, write a program to find the number of balanced strings for the strings of the length n.

For example, if $n = 3$, four strings, 010, 011, 100, 101, are balanced strings.

Input
Your program is to read from standard input. The first line of input contains one integer n ($1 \leq n \leq 100{,}000$).

Output
Your program is to write to standard output. Print exactly one line. The line should contain B modulo 16769023 where B is the number of balanced strings of length n.

The following shows sample input and output for three test cases.

Sample Input 1	Output for the Sample Input 1
3	4

Sample Input 2	Output for the Sample Input 2
22	2048

Sample Input 3	Output for the Sample Input 3
101	393256

Problem B
Balanced String
제한 시간: 0.5 초

0과 1로 이루어진 이진 문자열 0101101은 0과 1의 개수의 차이가 1 이하이다. 뿐만 아니라, 첫번째 문자를 포함하는 모든 부분 문자열 0, 01, 010, 0101, 01011, 010110, 0101101 모두 0과 1의 개수의 차이가 1 이하이다.

이와 같이, 이진 문자열 중에서 첫번째 문자를 포함하는 모든 부분 문자열의 0과 1의 개수의 차이가 1이하인 문자열을 **균형잡힌 문자열**이라 부른다. 문자열 자체도 자신의 부분 문자열이다.

양의 정수 n이 주어질 때, 길이가 n인 이진 문자열 중에서 균형잡힌 문자열의 수를 구하는 프로그램을 작성하시오.

예를 들어, $n = 3$인 경우에는 010, 011, 100, 101 네 개의 문자열이 균형잡힌 문자열이다.

Input
입력은 표준입력을 사용한다. 첫 번째 줄에 양의 정수 n $(1 \le n \le 100{,}000)$이 주어진다.

Output
출력은 표준출력을 사용한다. 길이가 n인 이진 문자열 중에서 균형잡힌 문자열의 개수를 16769023로 나눈 나머지 값을 한 줄에 출력한다.

다음은 세 개의 테스트 데이터에 대한 입출력 예이다.

Sample Input 1	Output for the Sample Input 1
3	4

Sample Input 2	Output for the Sample Input 2
22	2048

Sample Input 3	Output for the Sample Input 3
101	393256

Problem C
Byte Coin
Time Limit: 0.5 Seconds

International Capital and Property Company (ICPC) is investing money to Byte Coin, a virtual currency invented by Dr. Kim. In reality, the price of Byte Coin is unpredictable. But in this problem, assume that the price of Byte Coin can be accurately estimated in advance.

We are given stock prices of n days from day 1 to day n, like Figure 1, and the initial amount W of real money. The red square in Figure 1 represents the price Byte Coin on each day. We assume that we can sell and buy Byte Coins on each day. For buying and selling Byte Coins, we cannot split a Byte Coin. We want to maximize the money that we have when selling all the coins we have on day n.

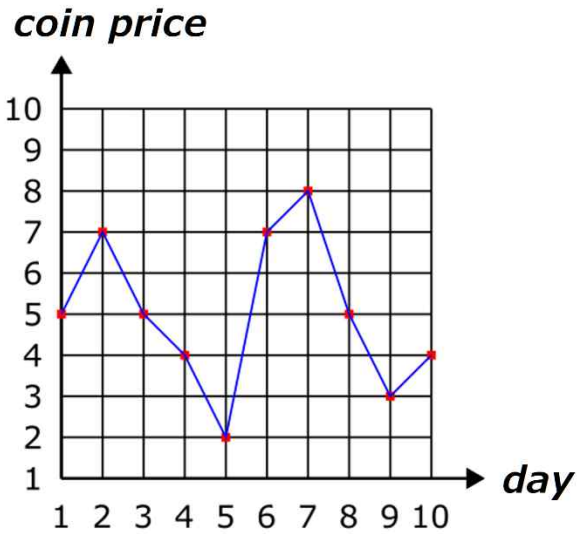


Figure 1. The Byte Coin price change for ten days

For example, the price of Byte Coin is given as Figure 1 in advance on the day 1 and the initial money W is 24. To maximize the revenue, the best strategy is like the following. On day 1, we buy 4 coins by paying money 20. On day 2, we sell all the coins and get money 28. So we have totally money 32. On day 5, we buy 16 coins by paying money 32. We sell all the coins on day 7 and have totally money 128. On day 9, we buy 42 coins by paying money 126 and on day 10, the last day, we sell all the coins. Then, we have money 170 on day 10, which is maximum.

Given the number of days n, the initial amount of money W, and the prices of Byte Coin on each of day 1 to n, write a program to maximize the final amount of money when selling all the coins we have on day n.

Input
Your program is to read from standard input. The input starts with a line containing two integers n and W ($1 \le n \le 15, 1 \le W \le 100{,}000$), where n is the number of days and W is the initial money. In the following n lines, the i-th line contains an integer s_i ($1 \le s_i \le 50$), where s_i denotes the price of Byte Coin on the day i.

Output

Your program is to write to standard output. Print exactly one line containing the maximum money that we have when selling all the coins we have on day n. Beware that even though the initial amount of money W is not so big but the final amount of money can be very large.

The following shows sample input and output for three test cases.

Sample Input 1

```
10 24
5
7
5
4
2
7
8
5
3
4
```

Output for the Sample Input 1

```
170
```

Sample Input 2

```
5 15
5
4
4
2
7
```

Output for the Sample Input 2

```
50
```

Sample Input 3

```
7 54
7
5
5
4
2
2
1
```

Output for the Sample Input 3

```
54
```

Problem C
Byte Coin
제한 시간: 0.5 초

국제자본부동산회사(ICPC)는 바이트 코인(Byte Coin)에 자금을 투자하고 있다. 바이트 코인은 김박사가 만든 가상 화폐이다. 실제로는 바이트 코인 가격을 예상할 수 없지만 이 문제에서는 바이트 코인 가격 등락을 미리 정확히 예측할 수 있다고 가정하자.

우리는 1일부터 n일까지 n일 동안 그림 1과 같이 바이트 코인의 등락을 미리 알 수 있으며 우리에게는 초기 현금 W가 주어져 있다. 그림 1의 빨간색 네모는 해당 일자의 바이트 코인 가격을 나타낸다. 매일 바이트 코인을 매수하거나 매도할 수 있다고 하자. 다만 바이트 코인 하나를 나누어 매도하거나 매수할 수는 없다. 우리는 n일 날 보유하고 있는 모든 코인을 매도할 때 가지고 있는 현금을 최대화하고 싶다.

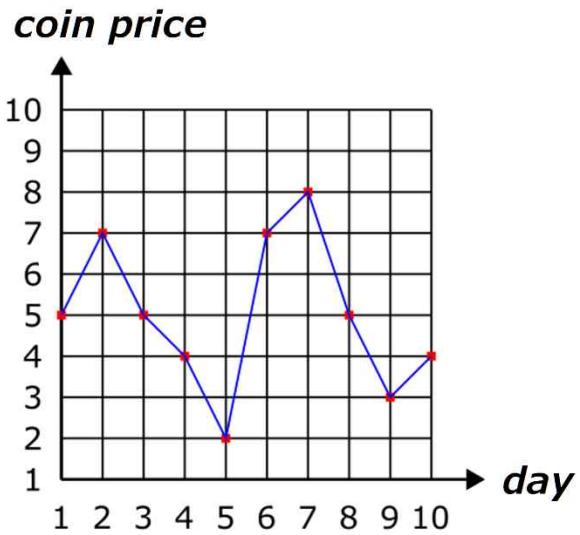


그림 1. 10 일간 바이트 코인 가격 등락 그래프

예를 들어, 그림 1과 같은 바이트 코인 등락 그래프를 첫날 미리 알 수 있다고 하고 우리에게 주어진 초기 현금이 24라고 하자. 수익을 최대한 높이려면 다음과 같이 바이트 코인을 매수, 매도할 수 있다. 첫 날 현금 20을 써서 4개의 코인을 산다. 둘째 날 모든 코인을 매도해서 현금 28을 얻고 모두 32의 현금을 갖게 된다. 5일째 되는 날 현금 32를 써서 16개의 코인을 매수한다. 7일째 되는

날 모든 코인을 매도해서 모두 128의 현금을 갖게 된다. 9일째 되는 날 현금 126을 써서 42개의 코인을 사고 10일 날 모든 코인을 매도한다. 그러면 10일 날 현금이 170이 되고 이것이 최대이다.

요일 수 n, 초기 현금 W, 1일부터 n일까지 각 요일의 바이트 코인 가격이 주어질 때, n일 날 보유하고 있는 모든 코인을 매도할 때 보유하게 될 최종 현금의 최댓값을 출력하는 프로그램을 작성하시오.

Input

입력은 표준입력을 사용한다. 첫 번째 줄에 요일 수를 나타내는 양의 정수 n과 초기 현금 $W(1 \leq n \leq 15, 1 \leq W \leq 100{,}000)$가 주어진다. 다음 n 개의 줄에서, i번째 줄은 i일의 바이트 코인 가격을 나타내는 정수 s_i가 주어진다$(1 \leq s_i \leq 50)$.

Output

출력은 표준출력을 사용한다. n일 날 보유하고 있는 모든 코인을 매도할 때 가지고 있는 현금의 최댓값을 한 행에 출력한다. 비록 초기 현금 W는 그렇게 크지 않지만 최종 현금은 매우 커질 수 있음에 주의하자.

다음은 세 개의 테스트 입출력 예이다.

Sample Input 1

```
10 24
5
7
5
4
2
7
8
5
3
4
```

Output for the Sample Input 1

```
170
```

Sample Input 2

```
5 15
5
4
4
2
7
```

Output for the Sample Input 2

```
50
```

Sample Input 3

```
7 54
7
5
5
4
2
```

Output for the Sample Input 3

```
54
```

2 1	

Problem D
Canal
Time Limit: 1.5 Seconds

It is quite recent that more people started to settle in Aissippissi, a small town located in a flat and dry desert. Aissippissi is still under development, and the government planned to construct a canal across the town. The canal will consist of two long and narrow waterways, stretching north-to-south and east-to-west, respectively, and crossing at some point. For the residents in Aissippissi, the canal will be very helpful for many purposes, so easy access to the canal is a very important criterion in its design.

The issue is where to put the canal, as the distance to the canal varies a lot for the residents depending on locations of the canal. You came up with an idea to resolve the issue: you want to locate the canal of two waterways such that the maximum distance to the canal over all houses is minimized.

Given the coordinates of n houses in Aissippissi, write a program that computes the best canal location that minimizes the maxmium distance to the canal from all the n houses and outputs the corresponding distance.

You may assume that n houses are points lying in the plane with the x- and y-axes, so that the x-axis is along the west-to-east direction and the y-axis along the south-to-north direction. Hence, the two waterways of the canal are regarded as two straight lines in the plane, being parallel to the x-axis and to the y-axis, respectively. The crossing point can be any point in the plane. In addition, the waterways can pass through some houses as they will be built a bit below the ground level. The distance from a house to the canal is the direct distance to the closer waterway of the canal on the plane. Hence, the distance from any house exactly on either waterway to the canal should be zero.

Input
Your program is to read from standard input. The input starts with a line containing an integer n ($1 \leq n \leq 300{,}000$), where n is the number of houses in Aissippissi. In the following n lines, each line contains two integers, which range inclusively from -10^9 to 10^9, representing the x- and y-coordinates, respectively, of each of the n houses. Note that there may be two or more houses with the same coordinates.

Output
Your program is to write to standard output. Print exactly one line consisting of a real number, rounded to the first decimal place, that represents the maximum distance from the houses to the best canal which minimizes the maximum distance.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
5 0 0 0 23 34 0 1 -21 -74 1	0.5

Sample Input 2	Output for the Sample Input 2
10 0 2 3 0 -10 0 0 23 0 -162 0 11 -235 0 -213 0 32 0 0 732	0.0

Problem E
Choreography
Time Limit: 0.5 Seconds

Korean idol groups are famous in the world for dynamic and elaborate group dances, called Kalgunmu. A Korean idol group has m members. One day the members practice their choreography in a stage. Suppose that the stage is simplified as a line. There are n closed intervals of identical length on the stage. Here we shall specify seven rules which the members should follow in every moment of choreography as follows:

R1) Each member should stand on one of the intervals.

R2) No two or more members can stand on the same interval.

R3) Two intervals on which arbitrary two members stand respectively cannot intersect.

Note that two closed intervals $[a, b]$ and $[c, d]$ with $a \leq c$ intersect if $c \leq b$.

The choreography consists of several steps with the following rules:

R4) At any step, only one member can move from her/his current interval to another interval. In other words, two or more members cannot move simultaneously at any step.

R5) When changing the interval on which she/he stands at some step, a member has to move into one of the intervals that her/his current interval intersects.

Note that a member cannot move into the interval I if I intersects an interval on which another member stands, according to the rule R3.

R6) The set of intervals on which the members stand in the start of choreography should be S.

R7) The set of intervals on which the members stand in the end of choreography should be E.

Note that the sets S and E are given as in the input.

According to the rules of the choreography, the size of S (or E) is m and the intervals in S (or E) do not intersect each other. Also each member shall move from an interval s in S to an interval e in E through the choreography. We are interested in the minimum number of steps to complete the choreography satisfying the above rules R1 to R7.

For example, in Figure E. 1, there are six intervals on a stage, which are denoted by numbers from 1 to 6. The blue intervals 1 and 4 are intervals on which two members stand in the start of choreography and the red intervals 3 and 6 are ones in the end.

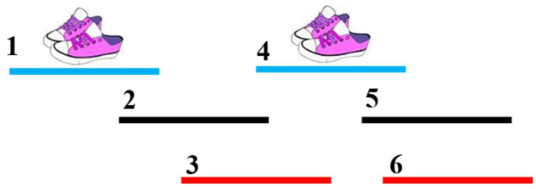


Figure E. 1 Initial configuration S of the choreography on a stage

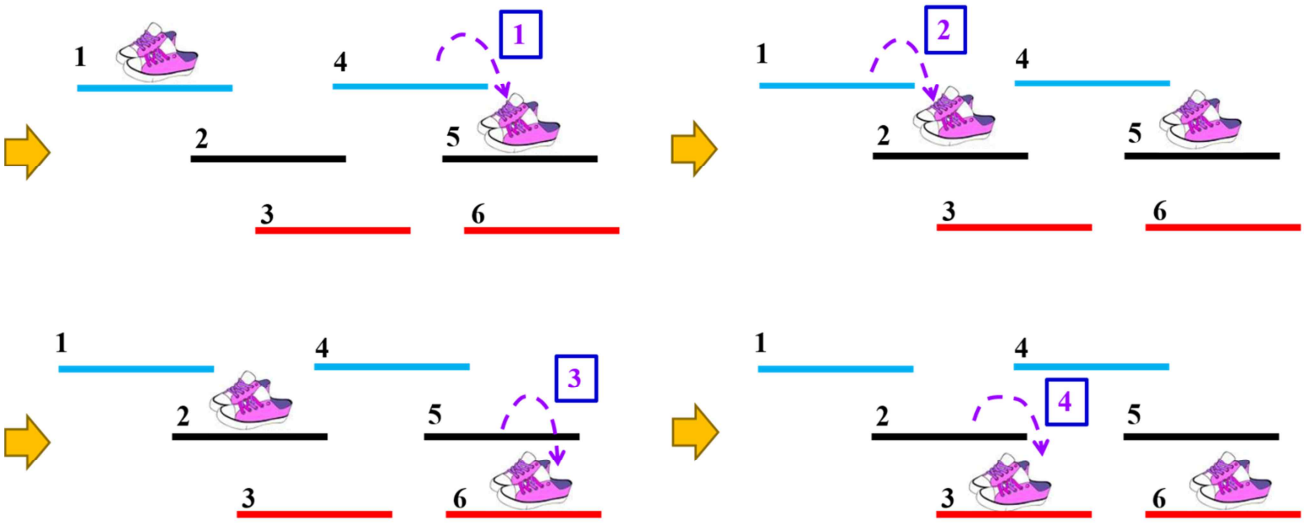


Figure E. 2 Four steps of choreography

At the first step, the member standing on the interval 4 moves into the interval 5. Notice that the member on the interval 1 cannot move into the interval 2 at the first step, because the interval 2 intersects the interval 4 that another member stands on. As in Figure E. 2, two members can move into the red intervals in four steps. But if the member on the interval 4 moves into the interval 6 at the first step, then the choreography would be completed in three steps, which is minimum.

Given m members, n intervals, and two sets S and E for the choreography, write a program to find a choreography from S to E with a minimum number of steps, and output all such steps.

Input

Your program is to read from standard input. The input starts with a line containing two integers, m and n ($1 \leq m \leq n \leq 5{,}000$), where m is the number of members and n is the number of intervals on the stage. The intervals are numbered from 1 to n such that the left endpoint of interval i is strictly less than the left one of interval $i+1$. The second line contains an integer r, the length of intervals ($1 \leq r \leq 10{,}000$). The third line contains a sequence of sorted n integers in ascending order where the i-th number is the left endpoint of the interval i, where these numbers are between 0 and 10^9. The following line contains a sequence of sorted m integers to denote the intervals belonging to the set S in ascending order. The last line contains a sequence of sorted m integers to denote the intervals belonging to the set E in ascending order.

Output

Your program is to write to standard output. The first line should contain the minimum number MIN of steps to complete the choreography. In the following MIN lines, the i-th line contains two integers a and b, where a member moves from the interval a to the interval b at the step i of the choreography with the minimum number of steps MIN. If there are one or more solutions, then print arbitrary one. If there is no solution, then print -1.

The following shows sample input and output for three test cases.

Sample Input 1

```
2 6
6
1 6 8 11 15 16
1 4
3 6
```

Output for the Sample Input 1

```
3
4 6
1 2
2 3
```

Sample Input 2

```
2 6
6
1 6 8 11 15 16
1 4
1 4
```

Output for the Sample Input 2

```
0
```

Sample Input 3

```
2 6
3
1 6 8 11 15 16
1 4
3 6
```

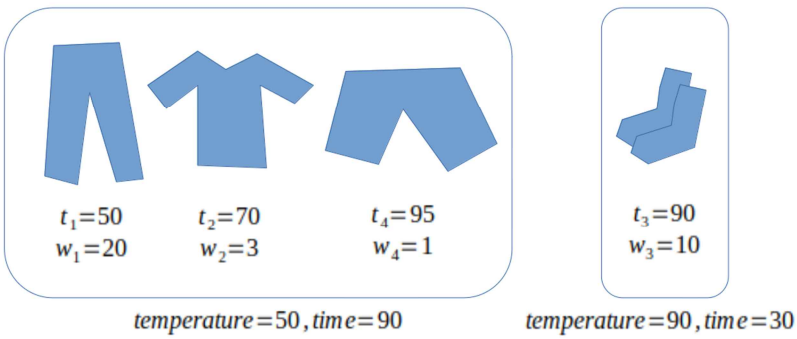
Output for the Sample Input 3

```
-1
```

Problem F
Dryer
Time Limit: 1 Second

You have n wet clothes you just pulled off the washer. You also have an electronic dryer. The dryer is large enough to dry all the clothes in one run. You can control the temperature of the drying air. However, if you dry all the clothes at a high temperature, some clothes will be damaged. Precisely, let t_i denote the highest temperature at which the i-th cloth can be dried without damage and let w_i denote the wetness of the i-th cloth. If you dry this cloth at the temperature T (of course, $T \leq t_i$), it will take $m_i = 30 + (t_i - T)\, w_i$ minutes. If you dry two or more clothes at once, the time the dryer takes is the longest m_i of these clothes. You should dry all the clothes without damage.

Because the dryer uses a lot of electricity, you are going to partition n clothes into at most k groups and runs the dryer for each group. Given n clothes with t_i's and w_i's, write a program to find the minimum total time to dry all the clothes without damage.



This figure illustrates an example of $n = 4, k = 2$. In this case, the total time is $90 + 30 = 120$ minutes.

Input
Your program is to read from standard input. The first line contains two integers n ($1 \leq n \leq 1{,}000$) and k ($1 \leq k \leq 3$). In the following n lines, the i-th line has two integers t_i ($40 \leq t_i \leq 100$) and w_i ($0 \leq w_i \leq 100$).

Output
Your program is to write to standard output. Print exactly one line containing the minimum total time as an integer.

The following shows sample input and output for three test cases.

Sample Input 1	Output for the Sample Input 1
4 2 50 20 70 3 90 10 95 1	120

Sample Input 2	Output for the Sample Input 2
4 2 50 20 70 3 90 1 95 1	85

Sample Input 3	Output for the Sample Input 3
1 3 40 4	30

Problem G
Enumeration
Time Limit: 1 Second

We are given a set $\sum$ of n English lowercase characters. We select k characters from $\sum$ without repetition and arrange these k characters in alphabetical order, then we get a word of k characters, which is called a *k-word*. For example, let $n = 5$, $k = 3$, and $\sum = \{a, b, c, d, e\}$. Then there are ten 3-words which are *abc*, *abd*, *abe*, *acd*, *ace*, *ade*, *bcd*, *bce*, *bde*, and *cde*. Given two distinct k-words, S and T, we want to enumerate all k-words satisfying two conditions: (C1) the first k-word is S and the last k-word is T, and (C2) the number of common characters in any two consecutive k-words is exactly $k - 1$. In the above example, if we enumerate all 3-words for $S = abd$ and $T = bde$, then we have a list of 3-words, (*abd*, *abe*, *abc*, *ace*, *bce*, *bcd*, *cde*, *acd*, *ade*, *bde*).

Given $\sum$, k, n, S, and T, enumerate all k-words so that the above two conditions (C1) and (C2) are satisfied.

Input
Your program is to read from standard input. The input consists of three lines. The first line contains two integers, n and k, where $2 \le n \le 20$ and $1 \le k \le n - 1$. The second line contains a string of n characters of $\sum$ in alphabetical order. The third line contains two distinct k-words S and T separated by a single space.

Output
Your program is to read from standard output. The first line contains an integer representing the number of k-words in the enumeration list which satisfies two conditions (C1) and (C2). The second line contains all k-words in the order of enumeration. If there are many solutions, print any one of the solutions. If there is no solution, print `-1` only.

The following shows sample input and output for three test cases.

Sample Input 1

```
5 3
abcde
abd bde
```

Output for the Sample Input 1

```
10
abd abe abc ace bce bcd cde acd
ade bde
```

Sample Input 2

```
5 1
abcde
d c
```

Output for the Sample Input 2

```
5
d a b e c
```

Sample Input 3

```
4 3
befy
efy bef
```

Output for the Sample Input 3

```
4
efy bey bfy bef
```

Problem H
Four Squares
Time Limit: 0.5 Seconds

It was proven by Lagrange in 1770 that every natural number can be represented as the sum of four or fewer squares. Some numbers are represented in multiple ways. For example, 26 is the sum of 5^2 and 1^2; it can also be represented as $4^2 + 3^2 + 1^2$. Expressing a number as the sum of four or fewer squares is historically a common problem posed to lightning calculators. It was reported in the early 1900s that a calculator produced a solution of $15663 = 125^2 + 6^2 + 1^2 + 1^2$ in 8 seconds. A more difficult problem took 56 seconds: $11339 = 105^2 + 15^2 + 8^2 + 5^2$.

Given a natural number n, write a program to express n as the sum of as few squares as possible.

Input
Your program is to read from standard input. The input consists of a single line containing a natural number n, where $1 \leq n \leq 50{,}000$.

Output
Your program is to write to standard output. Print exactly one line which contains the minimum number of squares whose sum is equal to n.

The following shows sample input and output for four test cases.

Sample Input 1	Output for the Sample Input 1
25	1

Sample Input 2	Output for the Sample Input 2
26	2

Sample Input 3	Output for the Sample Input 3
11339	3

Sample Input 4	Output for the Sample Input 4
34567	4

Problem H
Four Squares
제한 시간: 0.5 초

라그랑주는 1770 년에 모든 자연수는 넷 혹은 그 이하의 제곱수의 합으로 표현할 수 있다고 증명하였다. 어떤 자연수는 복수의 방법으로 표현된다. 예를 들면, 26은 5^2과 1^2의 합이다; 또한 $4^2 + 3^2 + 1^2$으로 표현할 수도 있다. 역사적으로 암산의 명수들에게 공통적으로 주어지는 문제가 바로 자연수를 넷 혹은 그 이하의 제곱수 합으로 나타내라는 것이었다. 1900 년대 초반에 한 암산가가 $15663 = 125^2 + 6^2 + 1^2 + 1^2$라는 해를 구하는데 8 초가 걸렸다는 보고가 있다. 좀 더 어려운 문제에 대해서는 56 초가 걸렸다: $11339 = 105^2 + 15^2 + 8^2 + 5^2$.

자연수 n이 주어질 때, n을 최소 개수의 제곱수 합으로 표현하는 컴퓨터 프로그램을 작성하시오.

Input
입력은 표준입력을 사용한다. 입력은 자연수 n을 포함하는 한 줄로 구성된다. 여기서, $1 \leq n \leq 50{,}000$이다.

Output
출력은 표준출력을 사용한다. 합이 n과 같게 되는 제곱수들의 최소 개수를 한 줄에 출력한다.

다음은 네 테스트 케이스에 대한 입출력 예이다.

Sample Input 1	Output for the Sample Input 1
25	1

Sample Input 2	Output for the Sample Input 2
26	2

Sample Input 3	Output for the Sample Input 3
11339	3

Sample Input 4	Output for the Sample Input 4
34567	4

Problem I
Registration
Time Limit: 1 Second

Print out your team's DOMJudge account ID and password.

Input
No input is given for this problem.

Output
Your program is to write to standard output. Print exactly two lines. The first line should contain your DOMJudge account ID, and the second line should contain your DOMJudge account password.

The following shows sample input and output, where the account ID is `team123` and the password is `passw0rd`. Notice that no input is given.

Sample Input	Output for the Sample Input
	`team123` `passw0rd`

Problem I
Registration
제한 시간: 1 초

자신이 속해 있는 팀의 DOMJudge ID 와 패스워드를 그대로 출력하는 프로그램을 작성하시오.

Input
이 문제는 입력이 없다.

Output
표준출력(standard output)으로 출력해야 한다. 첫 줄에 DOMJudge ID, 둘째 줄에 패스워드를 출력한다.

다음은 ID 가 `team123` 번, 패스워드가 `passw0rd` 인 경우의 입출력 예제이다. 참고로 입력이 없는 것에 주의한다.

Sample Input	Output for the Sample Input
	`team123` `passw0rd`

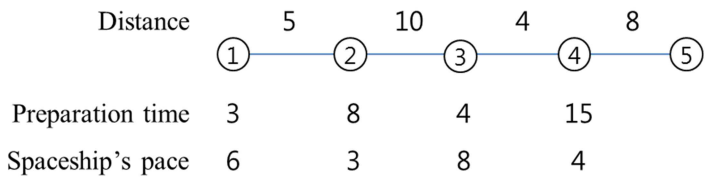
Problem J
Star Trek
Time Limit: 1 Second

The United Federation of Planets (abbreviated as UFP) consisting of n planets is an interstellar union of planetary governments. All planets are numbered from 1 to n. The headquarters of UFP is located on the planet numbered 1. UFP has established a linear interstellar path that connects in sequence from planet 1 to planet n.

The spaceships developed for interstellar travel have almost unlimited energy, allowing the ship to move without stopping along the travel route between two planets. Each planet has its own spaceship model. The performance of the spaceships is almost same, but the speed varies by model. The spaceship's pace is expressed as the number of hours required to move one light-year distance. The light-year is a distance unit used to express astronomical distances. If a ship's pace is 10, then it takes 10 hours to travel one light-year.

You, a resident on planet 1, are about to travel to planet n. In order to arrive at planet n as quickly as possible, you can transfer to spaceships at other planets on the way. However, you have to know that such a transfer needs additional time to prepare for landing, takeoff, entry and departure formalities, etc.

For example, suppose that there are five planets in UFP, given with the distances between adjacent planets, the spaceship's paces, and preparation times for the planets as in the figure below.



If a spaceship on planet 1 moves directly to planet 5, it takes 165 ($= 3 + 27 \times 6$) hours. If you transfer at planet 2, it takes 107 ($= 3 + 5 \times 6 + 8 + 22 \times 3$) hours. If you transfer at planet 2 and then at planet 4, it takes 130 ($= 3 + 5 \times 6 + 8 + 14 \times 3 + 15 + 8 \times 4$) hours. Considering all the possible travel plans, the minimum time to get to planet 5 is 107 hours.

Given the information of planets and spaceships of UFP, write a program to find the minimum time to get from planet 1 to planet n.

Input
Your program is to read from standard input. The input starts with a line containing an integer, n ($3 \le n \le 100{,}000$), where n is the number of planets of UFP. The planets are numbered from 1 to n. The next line consists of $n - 1$ integers, where the i-th integer represents the distance between planet i and planet $i + 1$. All distances are between 1 and 1,000. In the following $n - 1$ lines, the i-th line contains two integers, p and s ($0 \le p \le 10^9, 1 \le s \le 10^5$), where p is the preparation time and s is the spaceship's pace of planet i.

Output

Your program is to write to standard output. Print exactly one line. The line should contain an integer which represents the minimum time to travel from planet 1 to planet n.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
5 5 10 4 8 3 6 8 3 4 8 15 4	107

Sample Input 2	Output for the Sample Input 2
4 10 10 10 0 5 10 3 5 2	115

Problem K
Steel Slicing
Time Limit: 2.5 Seconds

ISCO(ICPC Steel Company) is a company that buys steel sheets of a certain shape, cuts them into pieces, and sells them in the industry market. Every steel sheet that ISCO buys is a polygon without holes such that each side is either horizontal or vertical with respect to the x-axis. The length of each side is a positive integer. We call such a polygon a *histogon*. See Figure 1(a) for a histogon.

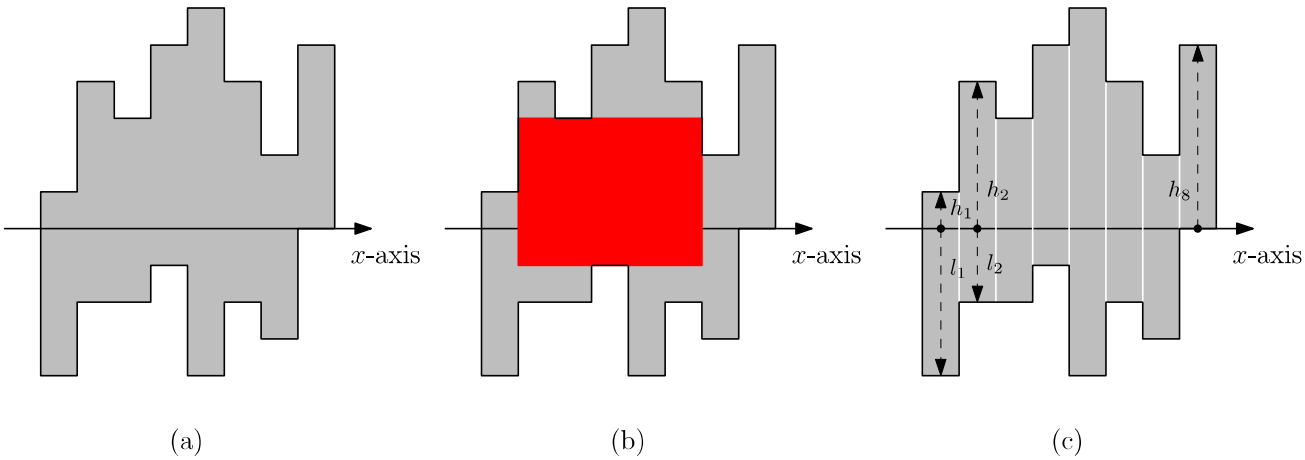


Figure 1. (a) A steel sheet which is a histogon. (b) An axis-aligned rectangle (red) with maximum area contained in the histogon. (c) The histogon is subdivided along vertical lines into 8 rectangular slabs with width 1.

Since the market price of a piece becomes much higher if the piece is a rectangle with larger area, it is desirable to cut a steel sheet so as to get a rectangle of maximum area. Thus, the task is to find a rectangle contained in a steel sheet with largest area. We assume that the rectangle should be axis-aligned, that is, every side is horizontal or vertical with respect to the x-axis. Formally, this problem can be stated as follows. Given a histogon, find an axis-aligned rectangle with maximum area contained in the histogon. Figure 1(b) shows an axis-aligned rectangle with maximum area that is contained in the histogon in Figure 1(a).

A histogon with width n can be subdivided along vertical lines into n rectangular slabs with width 1. The histogon in Figure 1(a) can be subdivided along vertical lines into 8 rectangular slabs, as shown in Figure 1(c). These slabs are numbered from 1 to n in order along the x-axis such that the leftmost slab has number 1. To ease the description, we assume that the x-axis intersects every slab. The x-axis intersects every slab in Figure 1(c). Then, slab i can be represented by two values, h_i and l_i, where h_i denotes the vertical length of slab i lying above the x-axis and l_i denotes the vertical length of slab i lying below the x-axis.

Given a histogon with width n, write a program to output the maximum area among the axis-aligned rectangles contained in the histogon.

Input

Your program is to read from standard input. The input starts with a line containing one integer n ($1 \leq n \leq 200{,}000$), where n is the width of the input histogon. The slabs are numbered from 1 to n such that the leftmost slab has number 1. In the following n lines, the i-th line contains 2 nonnegative integers that represent h_i and l_i ($0 \leq h_i, l_i \leq 1{,}000{,}000{,}000$).

Output

Your program is to write to standard output. Print exactly one line. The line should contain the maximum area among rectilinear rectangles contained in the given histogon.

The following shows sample input and output for two cases.

Sample Input 1	Output for the Sample Input 1
8 1 4 4 2 3 2 5 1 6 4 4 2 2 3 5 0	20

Sample Input 2	Output for the Sample Input 2
5 23 15 23 17 3 22 15 3 5 1	76

Problem L
Two Machines
Time Limit: 0.5 Seconds

A scheduling company SOPT has tasks $t_1, t_2, \ldots, t_n$ to complete. The company has two super machines A and B. To complete a task t_i, SOPT can choose only one of A and B. It takes a_i and b_i hours to complete the task t_i on the machines A and B, respectively. A machine can run at most one task at any time, and if it starts to run a task t_i then it cannot run another task t_j until the machine completes t_i. SOPT wants to minimize the completion time for all tasks.

For example, we have three tasks t_1, t_2, and t_3 with $a_1 = 2, b_1 = 3, a_2 = 5, b_2 = 3, a_3 = 2$, and $b_3 = 7$. The best way to minimize the completion time is to assign two tasks t_1 and t_3 to the machine A and to assign the other task t_2 to the machine B. Then A needs $2 + 2 = 4$ hours to complete t_1 and t_3, and B needs 3 hours to complete t_2, so the minimum completion time is 4 hours.

Given n tasks and the times to complete the tasks on machines A and B, write a program to output the minimum completion time to complete all tasks.

Input
Your program is to read from standard input. The input starts with a line containing one integer, n ($1 \le n \le 250$), where n is the number of tasks. In the following n lines, the i-th line contains two integers a_i and $b_i (1 \le a_i, b_i \le 250)$ where a_i and b_i denote the time to complete the task t_i on the machines A and B, respectively.

Output
Your program is to write to standard output. Print exactly one line. The line should contain the minimum completion time to complete all tasks.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
3 2 3 5 3 2 7	4

Sample Input 2	Output for the Sample Input 2
3 9 2 10 4 5 2	6

Problem L
Two Machines

제한 시간: 0.5 초

스케줄링 최적화 회사인 SOPT 에 완료해야 할 n개의 작업 $t_1, t_2, \ldots, t_n$ 이 있다. SOPT 회사는 두 대의 머신 A 와 B 를 보유하고 있다. 각 작업 t_i를 완료하기 위해 SOPT 는 머신 A 와 B 둘 중에 오직 하나를 선택할 수 있다. 작업 t_i를 완료하기 위해 머신 A를 선택하면 a_i시간이 걸리고 머신 B를 선택하면 b_i 시간이 걸린다. 각 머신은 어느 순간에 최대 하나의 작업만 수행할 수 있으며, 한 작업이 시작되면 그 작업을 완료하기 전까지 다른 작업을 그 머신에서 수행할 수 없다. SOPT 는 모든 작업을 완료하기 위한 최소의 완료 시간을 구하고자 한다.

예를 들어, 세 개의 작업이 t_1, t_2, t_3가 주어져 있고 $a_1 = 2, b_1 = 3, a_2 = 5, b_2 = 3, a_3 = 2, b_3 = 7$라고 하자. 완료 시간을 최소화하기 위해서는 작업 t_1, t_3는 머신 A에, 작업 t_2는 머신 B에 할당한다. 머신 A는 작업 t_1, t_3를 완료하는데 $2 + 2 = 4$ 시간이 걸리고 머신 B는 작업 t_2를 완료하는데 3 시간이 걸린다. 따라서 최소 완료 시간은 4 시간이 된다.

n개의 작업 $t_1, t_2, \ldots, t_n$ 과 각 머신에서 각 작업들을 수행하는 데 걸리는 시간들이 주어질 때, 모든 작업들을 완료하기 위해 걸리는 시간의 최솟값을 구하는 프로그램을 작성하시오.

Input

입력은 표준입력을 사용한다. 첫 번째 줄에 작업의 개수를 나타내는 양의 정수 n $(1 \le n \le 250)$이 주어진다. 다음 n개의 줄에서 i번째 줄에는 두 개의 정수 a_i, b_i $(1 \le a_i, b_i \le 250)$가 주어진다. 여기서 a_i와 b_i는 각각 작업 t_i를 머신 A와 B에서 완료하는데 걸리는 시간이다.

Output

출력은 표준출력을 사용한다. 모든 작업 $t_1, t_2, \ldots, t_n$을 완료하기 위한 최소의 완료시간을 한 줄에 출력한다.

다음은 두 가지 테스트 케이스에 대한 입출력 예이다.

Sample Input 1	Output for the Sample Input 1
3 2 3 5 3 2 7	4

Sample Input 2	Output for the Sample Input 2
3 9 2 10 4 5 2	6