| Problem | WEDDING | POWER |
|---|---|---|
| **Executable file** | WEDDING.EXE | POWER.EXE |
| **Source file** | WEDDING.PAS<br>WEDDING.C<br>WEDDING.CPP | POWER.PAS<br>POWER.C<br>POWER.CPP |
| **Input file** | WEDDING.IN | POWER.IN |
| **Output file** | WEDDING.OUT | POWER.OUT |
| **Time limit per test** | 10 seconds | 10 seconds |
| **Number of tests** | 10 | 10 |
| **Points per test** | 5 | 5 |
| **Total points** | **50** | **50** |
| | **100** | |

# WEDDING

Slavko is organizing a wedding for his friend Mirko.

The special feature of the wedding will be the popular dance called "Train". All guests line up one behind the other, and everyone except the first person in the train puts his hands on shoulders of the person in front of him, and then they cheerfully dance through the hall, and sometimes even through the kitchen.

Slavko wants his train to look nice. If two people who are next to each other are of very different heights, then the train doesn't look nice.

Mirko's family is very conservative, and it **isn't allowed** that anywhere in the train **in front of** somebody from Mirko's family is some **younger** member of his family. The rest of the people can stand wherever they want.

Given are the heights of all the guests, and the order by age of Mirko's family. Write a program that will distribute the people in the train, so that not one member of Mirko's family has a younger member in front of him, and the **sum of the differences of the heights** between the neighbors in the train is **minimal**.

## Input data

In the first line of the input file are two integers N and K, divided by a comma, $1 \leq N \leq 10000$, $1 \leq K \leq 1000$, $K \leq N$, the total number of people on the wedding and the number of people from Mirko's family.

In the next N lines are an integer V, $1000 \leq V \leq 2200$, the height of each guest. The guests are marked with numbers 1 to N, and the first K guests are members of Mirko's family, sorted by age descending (so the guest number 1 is the oldest member of Mirko's family, and guest K is the youngest).

## Output data

Write the minimum sum of differences in the first line.

In the next N lines write the markings of the guests in order in which they appear in the optimal train, from the beginning to the end.

**Note**: solution doesn't have to be unique.

## Examples

**WEDDING.IN**

```
3 2
2000
1200
1500
```

**WEDDING.OUT**

```
800
1
3
2
```

**WEDDING.IN**

```
5 3
1900
1300
1500
1200
1600
```

**WEDDING.OUT**

```
1000
1
5
4
2
3
```

**WEDDING.IN**

```
6 3
1700
1900
1500
1800
1750
1300
```

**WEDDING.OUT**

```
800
1
5
4
2
3
6
```

# POWER

Dobrica get an interesting and well-paid job. Every morning he needs to turn off all the street lamps in his village. All the lamps are located on the same side of a straight road.

Dobrica is at the party every night until 5 AM and then he starts to turn off the lamps. In the beginning, he is standing besides one of them.

Every lamp has a light bulb of a defined power, and because Dobrica is ecologically conscious, he wants to turn off all the lamps in order that would minimize total energy spent.

Dobrica is moving at the speed of 1 m/s because he is very tired. Turning off a lamp does not take extra time because he can do it while passing by.

Write a program to calculate **minimal energy that needs to be spent** to turn all the lamps off for given locations of lamps, powers of light bulbs and Dobrica's starting position.

## Input data

First line of the input file contains integer N, $2 \leq N \leq 1000$, the number of lamps in the village.

Second line contains integer V, $1 \leq V \leq N$, number of the lamp at which Dobrica started.

Each of the following N lines contains data describing a lamp - two integers D and W separated by a single space character, $0 \leq D \leq 1000$, $0 \leq W \leq 1000$. D is distance of a lamp from the beginning of the village (expressed in meters), and W is power of a light bulb in that lamp, i.e. the amount of energy spent by that light bulb during one second. Lamps are given in sorted order.

## Output data

First and only line of the output file should contain one integer - minimal amount of spent energy.

**Note**: the solution will always be smaller than 1,000,000,000.

## Examples

**POWER.IN**

```
3
2
1 4
6 5
9 7
```

**POWER.OUT**

```
65
```

**POWER.IN**

```
4
3
2 2
5 8
6 1
8 7
```

**POWER.OUT**

```
56
```

**POWER.IN**

```
6
5
3 2
11 10
12 18
13 19
15 15
17 19
```

**POWER.OUT**

```
370
```