# APIO '07

## Asia-Pacific Informatics Olympiad

**12th May, 2007**

**Duration: 5 hours**

**3 questions**

**All questions should be attempted**

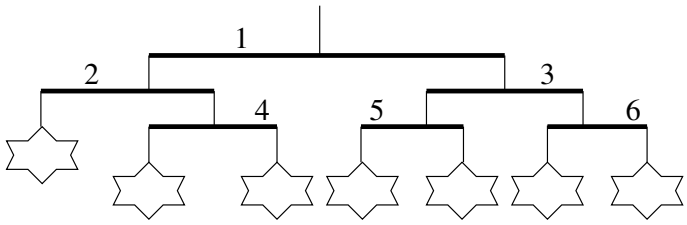
# Problem 1
# Mobiles

## Input File: *mobiles.in*
## Output File: *mobiles.out*

**Time and Memory Limits:** 1 second, 32 MB

You have been asked to buy a gift for your baby brother, Ike. However, you have noticed that Ike has a very particular taste in gifts. He only likes gifts that are configured in his particular style.

You have found a shop that sells mobiles. A *mobile* is a multi-layered decoration that is typically hung from the roof. Each mobile consists of a series of horizontal rods connected by vertical wires. Each rod has a wire hanging from both ends, which holds either another horizontal rod or a toy. A sample mobile is shown below:




To satisfy your brother, you need to find a mobile that can be reconfigured so that:

(i) any two toys are either at the same level (that is, joined to the roof by the same number of rods), or differ by only one level;

(ii) for any two toys that differ by one level, the toy to the left is further down than the toy to the right.

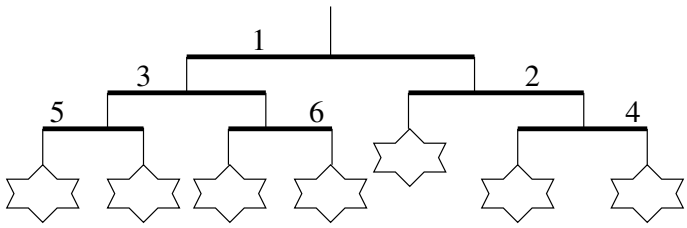
Mobiles can be reconfigured by performing *swaps*. A swap involves taking some rod, unhooking whatever is hanging beneath the left and right ends, and reattaching them at opposite ends (that is, the right and left ends respectively). This process does not modify the ordering of any rods or toys further down.

Since you are training for the Informatics Olympiad, you decide to write a program to test whether a given mobile can be reconfigured into a gift that Ike will like!

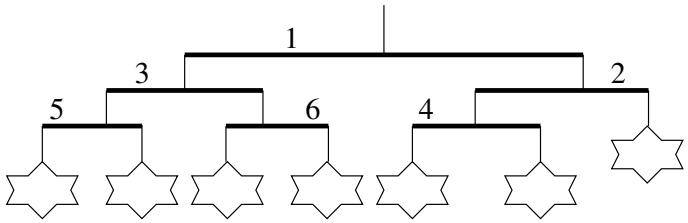
As an example, consider the mobile illustrated earlier. Ike will not like this mobile. Although it satisfies condition (i), it breaks condition (ii) — the toy at the leftmost end is at a higher level than the toys to its right.

However, the mobile *can* be reconfigured into a mobile that Ike will like. The following swaps are required:

1. First, the left and right ends of rod 1 are swapped. This exchanges the positions of rods 2 and 3, resulting in the following configuration:



2. Second, and finally, the left and right ends of rod 2 are swapped. This moves rod 4 to the left end of rod 2, and the toy to the right end of rod 2.




It can be seen that this final mobile satisfies Ike's requirements. All toys are at most one level apart, and the toys at a lower level are further to the left than the toys at a higher level.

Your task is, given a description of a mobile, to determine the smallest number of swaps required to reconfigure it so that Ike will like it (if this is possible). You may assume that the toys can never get in each other's way.

## Input

The first line of input will contain the single integer $n$ ($1 \leq n \leq 100\,000$) representing the number of rods in the mobile. The rods are numbered $1, 2, \ldots, n$.

The following $n$ lines will describe the connections for each rod. Specifically, the $i$th of these lines will describe rod $i$. Each of these lines will contain two integers $l$ $r$ separated by a single space, indicating what is hung beneath the left and right ends of the rod respectively. If a toy is hung beneath this rod, the corresponding integer $l$ or $r$ will be $-1$. Otherwise the integer $l$ or $r$ will be the number of a rod that is hung beneath this rod.

If there are any rods hanging beneath rod $i$, these rods will have numbers strictly greater than $i$. Rod 1 is the single rod at the top of the mobile.

## Output

Output should consist of a single line containing a single integer, giving the smallest number of swaps required to reconfigure the mobile according to Ike's constraints. If this is not possible, you should output the integer $-1$.

## Sample Input

```
6
2 3
-1 4
5 6
-1 -1
-1 -1
-1 -1
```

## Sample Output

```
2
```

## Explanation

The sample input above describes the first illustration in this problem statement.

## Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.

# Problem 2
# Backup

## Input File: *backup.in*
## Output File: *backup.out*

**Time and Memory Limits:** 1 second, 32 MB

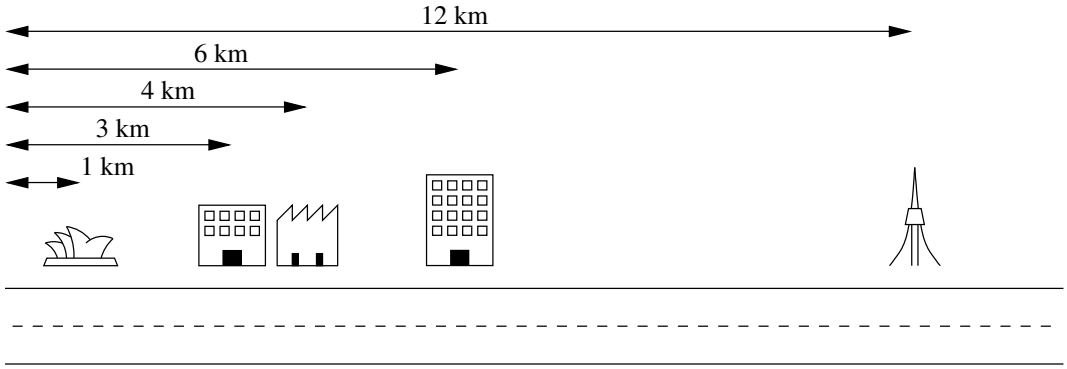
You run an IT company that backs up computer data for large offices. Backing up data is not fun, and so you design your system so that the different offices can back up each others' data while you sit at home and play computer games instead.

The offices are all situated along a single street. You decide to pair up the offices, and for each pair of offices you run a network cable between the two buildings so that they can back up each others' data.

However, network cables are expensive. Your local telecommunications company will only give you $k$ network cables, which means you can only arrange backups for $k$ pairs of offices ($2k$ offices in total). No office may belong to more than one pair (that is, these $2k$ offices must all be different).

Furthermore, the telecommunications company charges by the kilometre. This means that you need to choose these $k$ pairs of offices so that you use as little cable as possible. In other words, you need to choose the pairs so that, when the distances between the two offices in each pair are added together, the total distance is as small as possible.

As an example, suppose you had five clients with offices on a street as illustrated below. These offices are situated 1 km, 3 km, 4 km, 6 km and 12 km from the beginning of the street. The telecommunications company will only provide you with $k = 2$ cables.




The best pairing in this example is created by linking the first and second offices together, and linking the third and fourth offices together. This uses $k = 2$ cables as required, where the first cable has length 3 km – 1 km = 2 km, and the second cable has length 6 km – 4 km = 2 km. This pairing requires a total of 4 km of network cables, which is the smallest total possible.

### Input

The first line of input will contain the integers $n$ and $k$, representing the number of offices on the street ($2 \leq n \leq 100\,000$) and the number of available network cables ($1 \leq k \leq \frac{n}{2}$).

The following $n$ lines will each contain a single integer ($0 \leq s \leq 1\,000\,000\,000$), representing the distance of each office from the beginning of the street. These integers will appear in sorted order from smallest to largest. No two offices will share the same location.

## Output

Output should consist of a single positive integer, giving the smallest total length of network cable required to join $2k$ distinct offices into $k$ pairs.

## Sample Input

```
5 2
1
3
4
6
12
```

## Sample Output

```
4
```

## Explanation

The sample input above represents the example scenario described earlier.

## Scoring

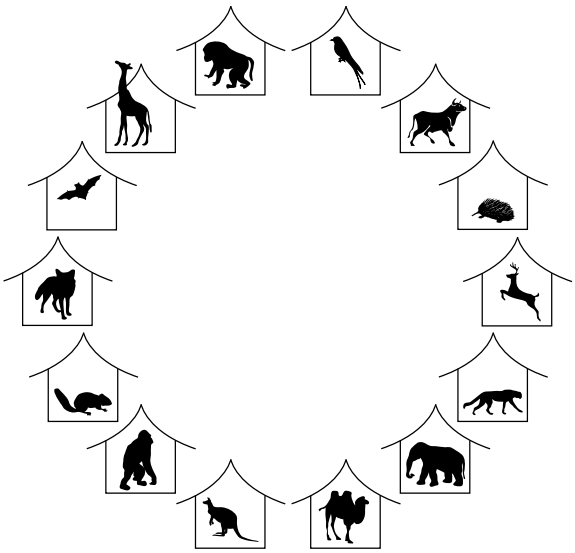
The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise. For 30% of the available marks, $n \leq 20$. For 60% of the available marks, $n \leq 10\,000$.

# Problem 3
# Zoo

## Input File: *zoo.in*
## Output File: *zoo.out*

**Time and Memory Limits:** 2 seconds, 16 MB

The pride of the Asia-Pacific region is the newly constructed Great Circular Zoo. Situated on a small Pacific island, it consists of a large circle of different enclosures, each containing its own exotic animal as illustrated below.



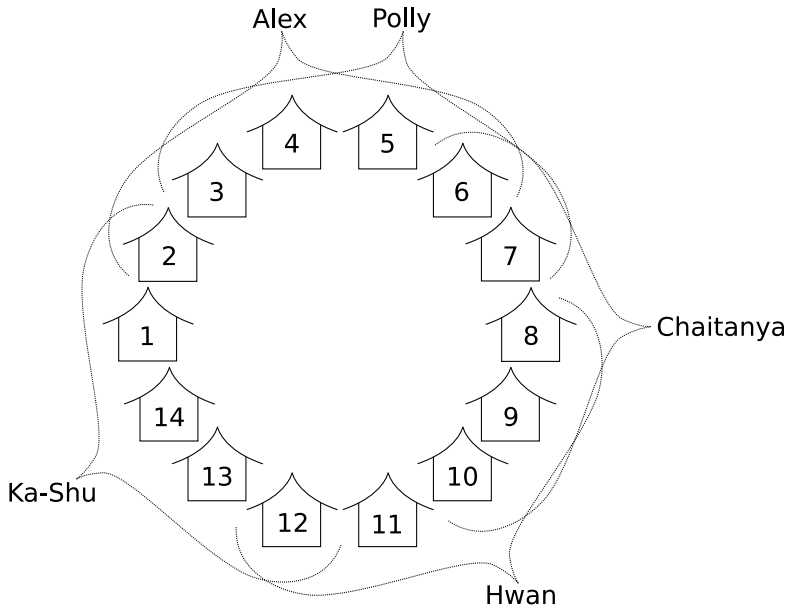
You are in charge of public relations for the zoo, which means it is your job to keep people as happy as possible. A busload of schoolchildren has just arrived, and you are eager to please them. However, this is no easy task—there are animals that some children love, and there are animals that some children fear. For example, little Alex loves monkeys and koalas because they are cute, but fears lions because of their sharp teeth. On the other hand, Polly loves lions because of their beautiful manes, but fears koalas because they are extremely smelly.

You have the option of removing some animals from their enclosures, so that children are not afraid. However, you are worried that if you remove too many animals then this will leave the children with nothing to look at. Your task is to decide which animals to remove so that as many children can be made happy as possible.

Each child is standing outside the circle, where they can see five consecutive enclosures. You have obtained a list of which animals each child fears, and which animals each child loves. A child will be made happy if either:

- At least one animal they fear is removed from their field of vision, **or**:
- At least one animal they love is *not* removed from their field of vision.

For example, consider the list of children and animals illustrated below:




| **Child** | **Enclosures Visible** | **Fears** | **Loves** |
|---|---|---|---|
| Alex | 2, 3, 4, 5, 6 | Enclosure 4 | Enclosures 2, 6 |
| Polly | 3, 4, 5, 6, 7 | Enclosure 6 | Enclosure 4 |
| Chaitanya | 6, 7, 8, 9, 10 | Enclosure 9 | Enclosures 6, 8 |
| Hwan | 8, 9, 10, 11, 12 | Enclosure 9 | Enclosure 12 |
| Ka-Shu | 12, 13, 14, 1, 2 | Enclosures 12, 13, 2 | — |

Suppose you remove the animals from enclosures 4 and 12. This will make Alex and Ka-Shu happy, because at least one animal that they fear has gone. This will also keep Chaitanya happy, since both enclosures 6 and 8 still contain animals that he loves. However, both Polly and Hwan will be unhappy, since they cannot see any animals that they love but they can still see all the animals that they fear. This arrangement therefore gives a total of three happy children.

Now suppose you put these animals back into their enclosures, and remove the animals from enclosures 4 and 6 instead. Alex and Polly will be happy because the animals that they fear in enclosures 4 and 6 have gone. Chaitanya will be happy because, even though animal 6 has gone, he can still see the animal in enclosure 8 which he loves. Likewise, Hwan will be happy because she can now see the animal in enclosure 12, which she loves. The only person unhappy will be Ka-Shu.

Finally, suppose you put the animals back once more and then remove only the animal from enclosure 13. Ka-Shu will now be happy since one animal that he fears has been removed, and Alex, Polly, Chaitanya and Hwan will all be happy since they can all see at least one animal that they love. Thus this arrangement gives five happy children, the largest number possible.

## Input

The first line of input will be of the form $N\ C$, where $N$ is the number of animal enclosures ($10 \leq N \leq 10\,000$) and $C$ is the number of children ($1 \leq C \leq 50\,000$). The enclosures are numbered $1, 2, \ldots, N$ clockwise around the circle.

Following this will be $C$ additional lines of input, where each line describes a single child. Each of these lines will be of the form

$$E\ F\ L\ X_1\ X_2\ \cdots\ X_F\ Y_1\ Y_2\ \cdots\ Y_L,$$

where:

- $E$ is the first enclosure that the child can see ($1 \leq E \leq N$). In other words, the child can see enclosures $E$, $E+1$, $E+2$, $E+3$ and $E+4$. Note that numbers larger than $N$ wrap back around the circle, so if $N = 14$ and $E = 13$ then the child can see enclosures 13, 14, 1, 2 and 3.
- $F$ is the number of animals that the child fears, and $L$ is the number of animals that the child loves.
- Enclosures $X_1, \ldots, X_F$ contain the animals that the child fears ($1 \leq X_1, \ldots, X_F \leq N$).
- Enclosures $Y_1, \ldots, Y_L$ contain the animals that the child loves ($1 \leq Y_1, \ldots, Y_L \leq N$).
- No two of the integers $X_1, \ldots, X_F, Y_1, \ldots, Y_L$ are equal, and all of these integers describe enclosures that the child can see.

Children will be listed in sorted order according to the first enclosure $E$ (so the child with lowest $E$ will appear first and the child with largest $E$ will appear last). Note that more than one child may have the same first enclosure $E$.

## Output

Output must consist of a single integer, giving the largest number of children that can be made happy.

## Sample Input 1

```
14 5
2 1 2 4 2 6
3 1 1 6 4
6 1 2 9 6 8
8 1 1 9 12
12 3 0 12 13 2
```

## Sample Output 1

```
5
```

## Sample Input 2

```
12 7
1 1 1 1 5
5 1 1 5 7
5 0 3 5 7 9
7 1 1 7 9
9 1 1 9 11
9 3 0 9 11 1
11 1 1 11 1
```

## Sample Output 2

```
6
```

## Explanation

The first sample case is the example discussed earlier, in which all $C = 5$ children can be made happy. The second sample case is an example in which it is impossible to make all $C = 7$ children happy.

## Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.