Programming Assignment 2 (Due Date: 10 April 2020)

For this programming assignment, you will not have to submit any written answers. Instead, you will have to submit a program written in Java or C++11 on CodeCrunch at `https://codecrunch.comp.nus.edu.sg/`. The portal will stop accepting submissions after 10 April 2020 2359h so please start your assignment **early**.

Templates will be provided for all the problems. These templates provide a starting point for your implementation. It is highly recommended to use all the templates given to you. **Do not change the file name or the class name of the template, or else your code may be marked as incorrect**. However, you have to submit your own work. **Posting the question or solution in public repositories are not allowed also counts as a form of plagiarism.** Any form of plagiarism is subject to disciplinary action.

Please include a brief description of your algorithm as a comment at the top of the code as this will be used for marking. You should also explain the time complexity of your algorithm in this comment. Task A is a greedy task. You should give a brief proof of your algorithm's correctness in the comment.

There are some example test cases provided in the assignment folder and these will also be uploaded onto CodeCrunch. If your program fails any of the example test cases, you will get **ZERO** marks. If your program passes all the example test cases, your code will be marked manually so please code neatly and add comments where appropriate. You may be asked to explain your code if the marker cannot understand it. Marks will be deducted if there are bugs in your code or if your algorithm does not meet the time complexity stated in the question. You are strongly encouraged to design your own test cases to test your code.

Note: Passing all the test cases on CodeCrunch does not guarantee that you will get full credit for the assignment.

Task A (3%)

Alice and Bob are learning a module named **Design and Analysis of Algorithms**. They are given a set of $2N$ problems for their programming assignment.
Since output of the previous problem is input of the next problem, the problems must be solved in a strict order. Formally, for every $1 \leq i < 2N$, problem i must be solved before problem $i + 1$.
Their parents know the friends very well, so they can estimate that for the i-th problem ($1 \leq i \leq 2N$), Bob will solve it in $b[i]$ minutes while it takes Alice $a[i]$ minutes to do the same. Bob and Alice want to cooperate to solve all the problems, each will solve N of them (for maintaining a good relationship).
Luckily, their parents agree with the plan, but they are not smart enough to distribute the problems in such a way that minimizes total number of minutes for Bob and Alice to solve all problems. And they know that you are very smart! Please help them to find a way to distribute problems in such a way that all problems can be solved in minimum number of minutes.

Input

The first line contains a positive integer N.
Next $2N$ lines, the i-th line contains two positive integers $a[i]$ and $b[i]$.

Output

A single line contains the smallest number of minutes for Bob and Alice to solve all problems.

Example Input

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

Example Output

```
8
```

Explanation

8 is the smallest number of minutes for Bob and Alice to solve all problems. Bob will solve the second and third problem, while Alice will solve the two remaining ones.

Limits

- $1 \leq N \leq 100000$
- $1 \leq a[i], b[i] \leq 1000$
- Your program should terminate within 1 second for C++11 and 2 seconds for Java
- Your algorithm should have a **worst case time complexity** of $O(N \log N)$

Task B1 (4%)

Alice have a pack of $N + 1$ **identical** balloons. She took one balloon and blow $M + 1$ cm^3 of air into it and the balloon pop.
Being curious, she wants to find the **elastic score** of those balloons(same for all balloons), define as the largest integer amount of air x in cm^3 such that a balloon can contain that amount of air without being pop. Note that, the elastic score of those balloons must be a non-negative integer less than or equal to M.
In order to do that, she will conduct trials. Each trial involve blowing x cm^3 of air into a balloon ($1 \leq x \leq M$) and either the balloon pop or not. If the balloon pop, it can not be used in later trials. Being lazy, she wants to know the answer with the least trials. You are her good friend so she ask you to find out the number of trials needed in the worst case.

Input

A single line contains two integers representing N and M.

Output

A single number represent the number of trials needed.

Example Input

```
2 6
```

Example Output

```
3
```

Explanation

First Alice will blow $3cm^3$ of air into a balloon.
If the balloon pop, there is only 1 balloon remain, which is used in 2 more trials with $1cm^3$ and $2cm^3$ in the worst case.
If the balloon doesn't pop, she can reuse the balloon in next trial with $5cm^3$. No matter if this balloon pop or not, she still have 1 balloon for the third trial with $4cm^3$ or $6cm^3$.

Limits

- $1 \leq N \leq 100, 1 \leq M \leq 200$
- Your program should terminate within 1 second for C++11 and 2 seconds for Java

- Your algorithm should have a **worst case time complexity** of $O(NM^2)$
- Hard-coded answer will result in **ZERO** score.

Task B2 (Challenge question) (1%)

Now that Alice has bought giant balloons, things are getting out of hand. Can you help her this time?

Input

A single line contains two integers representing N and M.

Output

A single number represent the number of trials needed.

Example Input

```
20 3230
```

Example Output

```
12
```

Limits

- $1 \leq N \leq 100, 1 \leq M \leq \mathbf{10^9}$
- Your program should terminate within 1 second for C++11 and 2 seconds for Java

Note

This question is graded on binary basis.