

题目描述：

在某个项目中有多个任务（用 $tasks$ 数组表示）需要您进行处理，其中 $tasks[i] = [s_i, e_i]$,

你可以在 $s_i \leq day \leq e_i$ 中的任意一天处理该任务。请返回你可以处理的最大任务数。

注：一天可以完成一个任务的处理。

输入描述：

第一行为任务数量 n , $1 \leq n \leq 100000$ 。后面 n 行表示各个任务的开始时间和终止

时间，用 s_i 和 e_i 表示， $1 \leq s_i \leq e_i \leq 100000$ 。

输出描述：

输出为一个整数，表示可以处理的最大任务数。

示例 1

输入：

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

输出：

```
3
```

说明：

第一天处理任务 1，第二天处理任务 2，第三天处理任务 3。

```
#include <algorithm>
#include <iostream>
#include <queue>
#include <tuple>
#include <vector>
using namespace std;

struct my_gt : public binary_function<pair<int, int>, pair<int, int>, bool> {
    bool operator()(const pair<int, int> &__x, const pair<int, int> &__y) const { return
__x.second > __y.second; }
};

int main() {
    int n;
    cin >> n;
```

```
int se[2 * n];
int max = 0;

for (int i = 0; i < n; i++) {
    cin >> se[2 * i];
    cin >> se[2 * i + 1];
    if (se[2 * i] > max) max = se[2 * i];
    if (se[2 * i + 1] > max) max = se[2 * i + 1];
}

bool used[max + 1];
fill(used, used + max + 1, 0);

int num = min(max, n);

cout << num << endl;
}
```