

6. Операционные системы lite

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Васин жесткий диск состоит из M секторов. Вася последовательно устанавливал на него различные операционные системы следующим методом: он создавал новый раздел диска из последовательных секторов, начиная с сектора номер a_i и до сектора b_i включительно, и устанавливал на него очередную систему. При этом, если очередной раздел хотя бы по одному сектору пересекается с каким-то ранее созданным разделом, то ранее созданный раздел «затирается», и операционная система, которая на него была установлена, больше не может быть загружена. Напишите программу, которая по информации о том, какие разделы на диске создавал Вася, определит, сколько в итоге работоспособных операционных систем установлено и работает в настоящий момент на Васином компьютере.

Формат ввода

Сначала вводятся натуральное число M — количество секторов на жестком диске ($1 \leq M \leq 10^9$) и целое число N — количество разделов, которое последовательно создавал Вася ($0 \leq N \leq 1000$). Далее идут N пар чисел a_i и b_i , задающих номера начального и конечного секторов раздела ($1 \leq a_i \leq b_i \leq M$).

Формат вывода

Выведите одно число — количество работающих операционных систем на Васином компьютере.

Пример 1

Ввод

Вывод

10

1

3

1 3

4 7

3 4

Пример 2

Ввод

Вывод

10

3

4

1 3

4 5

7 8

4 6

Язык

GNU GCC 12.2 C++20

Набрать здесь

Отправить файл

```
1 #include <iostream>
2 #include <set>
3 #include <utility>
4
5 int main() {
6     int M;
7     std::cin >> M;
8
9     int N;
10    std::cin >> N;
11}
```

```
12     std::set<std::pair<int, int>> v;  
13     std::pair<int, int> remove {-1, -1};  
14     for (int i = 0, tmp_s, tmp_e; i != N; ++i) {  
15         std::cin >> tmp_s >> tmp_e;  
16         if (v.empty()) {  
17             v.insert({tmp_s, tmp_e});  
18         } else {  
19             for (auto it = v.begin(); it != v.end(); ++it) {  
20                 if (remove.first != -1) {  
21                     v.erase(remove);  
22                     remove = {-1, -1};  
23                 }  
24                 if ((((*it).first <= tmp_s && tmp_s <= (*it).second))  
25                     || ((*it).first <= tmp_e && tmp_s <= (*it).second))  
26                     || ((tmp_s <= (*it).first && (*it).first <= tmp_e) || ((tmp_s <= (*it).second && (*it).second <=  
27                         remove=*it;  
28                     if (v.size() == 0) {  
29                         break;  
30                     }  
31                 }  
32             }  
33             if (remove.first != -1) {  
34                 v.erase(remove);  
35                 remove = {-1, -1};  
36             }  
37             v.insert({tmp_s, tmp_e});  
38         }  
39     }
```