

## Задача D. День рождения

Ограничение по времени: 0.2 секунды.

Ограничение по памяти: 64 мегабайта.

На день рождения пришли  $N$  человек. В некоторый момент именинник решил, что пора устроить какую-нибудь игру. Он выяснил, что  $i$ -й человек согласен вступить в игру, если в ней уже принимают участие не менее  $A_i$  и не более  $B_i$  человек. Единожды вступив в игру, никто из нее не выходит. Требуется выяснить, может ли именинник установить такую последовательность вступления в игру, что в итоге все присутствующие станут ее участниками. (Сам именинник в игре участия не принимает.)

### Формат входного файла

В первой строке задается целое число  $N$  ( $1 \leq N \leq 10^5$ ) - количество гостей. В каждой из последующих  $N$  строк содержится пара целых чисел  $A_i$  и  $B_i$  ( $0 \leq A_i \leq B_i < N$ ).

### Формат выходного файла

Если можно установить последовательность вступления гостей в игру, чтобы в итоге все стали ее участниками, то нужно вывести номера гостей в том порядке, в каком они могут вступать в игру. Если всех вовлечь в игру не удастся, выведите одно число 0.

### Примеры

Стандартный поток ввода	Стандартный поток вывода
5 4 4 0 3 1 4 1 3 2 2	2 3 5 4 1
3 1 1 1 1 1 1	0

### Анализ решения задачи

Задачу при таких ограничениях можно решить методом полного перебора – на каждом этапе выбирать того человека, который уже может вступить в игру ( $A_i$  меньше либо равно количеству уже вступивших в игру на данном этапе). Из тех, кто уже может вступить в игру, следует выбрать тех, кто имеет наименьшее  $B_i$  – те, кто готовы играть при большем количестве вступивших в игру еще могут подождать.

Таким образом, помечаем всех, кто уже вступил в игру. Следует сохранять порядок вхождения в игру в отдельный массив, чтоб легко можно было их вывести в выходной файл.

Если в итоге получилось упорядочить всех  $N$  человек – выводим их в порядке вхождения в игру, иначе – выводим 0. Если на каком-то этапе не получилось подобрать человека, готового войти на данном этапе в игру – прекращаем поиск и выводим итоговый результат.

## Пример решения

```
1 var n,i,j,k:integer;
2   r:array[1..100000,1..3] of integer;
3   t:array[1..100000] of integer;
4 begin
5   readln(n);
6   for i:=1 to n do
7     begin
8       readln(r[i,1],r[i,2]);
9       t[i]:=0; r[i,3]:=0;
10    end;
11  for i:=1 to n do
12    begin
13      k:=0;
14      for j:=1 to n do
15        if (r[j,3]=0) and (r[j,1]<=i-1) and (r[j,2]>=i-1) and ((k=0) or (r[k,2]>r[j,2])) then k:=j;
16      if k>0 then
17        begin
18          t[i]:=k;
19          r[k,3]:=1;
20        end
21      else Break;
22    end;
23  if t[n]>0 then
24    begin
25      write(t[1]);
26      for i:=2 to n do write(' ',t[i]);
27      writeln;
28    end
29  else Writeln(0);
30  readln;
31 end.
```

При больших объемах данных можно уменьшить сложность алгоритма, передвигая на каждом этапе уже вступившего в игру участника в начало списка в порядке их вхождения. Тогда для каждого следующего игрока не придется в цикле перебирать уже играющих людей изменив цикл строки 14 на примерно такой: **for j:=i to n ...**, причем в таком случае не нужно будет проверять на равенство нулю  $r[j,3]$ .