

题目描述：

有一个总空间为 100 字节的堆，现要从中新申请一块内存，内存分配原则为优先紧接着前一块已使用内存分配空间足够且最接近申请大小的空闲内存。

输入描述：

输入：

第 1 行是 1 个整数，表示期望申请的内存字节数；

第 2 到 N 行是用空格分割的两个整数，表示当前已分配的内存的情况，每一行表示一块已分配的连续内存空间，每行的第 1 和第 2 个整数分别表示偏移地址和内存块大小，如：

0 1

3 2

表示 0 偏移地址开始的 1 个字节和 3 偏移地址开始的 2 个字节已被分配，其余内存空闲。

输出描述：

输出：

若申请成功，输出申请到内存的偏移；若申请失败，输出-1。

补充说明：

1. 若输入信息不合法或无效，则申请失败。
2. 若没有足够的空间供分配，则申请失败。
3. 堆内存信息有区域重叠或有非法值等都是无效输入。

输入：

1

0 1

3 2

输出：

1

说明：

堆中已使用的两块内存是偏移从 0 开始 1 字节和偏移从 3 开始的 2 字节，空闲的两块内存是偏移从 1 开始 2 个字节和偏移从 5 开始 95 字节，根据分配原则，新申请的内存应从 1 开始分配 1 个字节，所以输出偏移为 1。

```
target = int(input().strip())
```

```
total_data = [i for i in range(100)]
```

```
flag = True
```

```
while True:
```

```
    try:
```

```
        cur_data = list(map(int, input().strip().split()))
```

```
    except EOFError:
```

```
        break
```

```
    if cur_data[0] < 0 or cur_data[0] >= 100:
```

```
        flag = False
```

```
    elif cur_data[1] <= 0 or cur_data[0] + cur_data[1] > 100:
```

```
        flag = False
```

```
    else:
        for i in range(cur_data[0], cur_data[0] + cur_data[1]):
            if total_data[i] == -1:
                flag = False
                total_data[i] = -1

if not flag:
    print(-1)
elif target <= 0:
    print(-1)
else:
    flag = True
    for i in range(100):
        if -1 not in total_data[i:i+target] and i + target <= 100:
            print(i)
            flag = False
            break
    if flag:
        print(-1)
```