

## 实验七

## 编程题 1 (总分: 100.00)

#	题目	分值
1	实验7-1: Lecture Halls	100.00

**Description**

设有 $n$ 个活动，每个活动都要求使用同一资源，而在同一时间内只有一个活动能使用这一资源。每个活动都有一个要求使用该资源的起始时间 $s_i$ 和一个结束时间 $f_i$ ，且 $s_i < f_i$ 。如果选择了活动 $i$ ，则它在半开区间 $[s_i, f_i)$ 内占用资源。若区间 $[s_i, f_i)$ 与区间 $[s_j, f_j)$ 不相交，则称活动 $i$ 与活动 $j$ 是相容的。也就是说，当 $s_i \geq f_j$ 或 $s_j \geq f_i$ 时，活动 $i$ 与活动 $j$ 相容。

编程任务：对于给定的 $k$ 个待安排的活动，编程计算安排最多的活动数。提示：需要排序

**Input**

输入数据是由多组测试数据组成。每组测试数据输入的第一行有1个正整数 $k$ ，表示有 $k$ 个待安排的活动。接下来的 $k$ 行中，每行有2个正整数，分别表示 $k$ 个待安排的活动开始时间和结束时间。时间从0点开始计算。

**Output**

对应每组输入，输出是计算出的最多可以安排的活动数。

**Sample Input**

```
5
1 23
12 28
25 35
27 80
36 50
```

## Sample Output

3

参考答案:

```
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    int start;
    int end;
} Activity;

// 比较函数用于qsort排序
int compare(const void *a, const void *b) {
    return ((Activity *)a)->end - ((Activity *)b)->end;
}

int activitySchedule(Activity activities[], int n) {
    qsort(activities, n, sizeof(Activity), compare);

    int count = 1; // 记录选择的活动数量
    int lastEnd = activities[0].end;

    for (int i = 1; i < n; i++) {
        if (activities[i].start >= lastEnd) {
            count++;
            lastEnd = activities[i].end;
        }
    }

    return count;
}

int main() {
    int n;
    scanf("%d", &n);

    Activity activities[n];

    for (int i = 0; i < n; i++) {
        scanf("%d %d", &activities[i].start, &activities[
i].end);
    }

    int result = activitySchedule(activities, n);
    printf("%d\n", result);

    return 0;
}
```

