

```

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

#define MAX_SIZE 100

// 比较函数，用于排序请求数组
int compare(const void *a, const void *b) {
    const int *req1 = (const int *)a;
    const int *req2 = (const int *)b;
    return (req1[1] - req2[1]); // 按照期望完成时间排序
}

int main() {
    int req[MAX_SIZE][3]; // 每个请求包含 [时间, 期望完成时间, 实际开始时间]
    int requestNum;
    int start = 0;
    int maxDelay = 0;

    printf("输入请求数量: ");
    scanf("%d", &requestNum);

    if (requestNum > MAX_SIZE) {
        printf("请求数量过多\n");
        return 0;
    }

    // 输入每个请求的时间和期望完成时间
    for (int i = 0; i < requestNum; ++i) {
        printf("请求 %d 的时间和期望完成时间: ", i + 1);
        scanf("%d %d", &req[i][0], &req[i][1]);
        req[i][2] = 0; // 初始化实际开始时间
    }

    // 按照期望完成时间排序请求
    qsort(req, requestNum, sizeof(req[0]), compare);

    // 计算实际开始时间、实际完成时间和最大延迟
    for (int i = 0; i < requestNum; ++i) {
        req[i][2] = start; // 实际开始时间
        start += req[i][0]; // 更新当前时间
        int end = start; // 实际完成时间
        if (maxDelay < end - req[i][1]) {
            maxDelay = end - req[i][1]; // 更新最大延迟
        }
    }

    printf("最小的最大延迟: %d\n", maxDelay);

    return 0;
}

```