#include <stdio.h>

#include <stdlib.h>

struct Job {

    int id;

    int deadline;

    int profit;

};

int compare(const void \*a, const void \*b) {

    struct Job \*jobA = (struct Job \*)a;

    struct Job \*jobB = (struct Job \*)b;

    return (jobB->profit - jobA->profit);

}

int findMaxDeadline(struct Job jobs[], int n) {

    int max = jobs[0].deadline;

    for (int i = 1; i < n; i++) {

        if (jobs[i].deadline > max)

            max = jobs[i].deadline;

    }

    return max;

}

void jobScheduling(struct Job jobs[], int n) {

    qsort(jobs, n, sizeof(struct Job), compare);

    int max\_deadline = findMaxDeadline(jobs, n);

    int result[max\_deadline + 1];

    for (int i = 0; i <= max\_deadline; i++)

        result[i] = -1;

    int jobCount = 0;

    int totalProfit = 0;

    for (int i = 0; i < n; i++) {

        for (int j = jobs[i].deadline; j > 0; j--) {

            if (result[j] == -1) {

                result[j] = jobs[i].id;

                jobCount++;

                totalProfit += jobs[i].profit;

                break;

            }

        }

    }

    printf("Job sequence that maximizes profit:\n");

    for (int i = 1; i <= max\_deadline; i++) {

        if (result[i] != -1)

            printf("Job %d\n", result[i]);

    }

    printf("Total number of jobs done: %d\n", jobCount);

    printf("Total profit: %d\n", totalProfit);

}

int main() {

    int n;

    printf("Enter the number of jobs: ");

    scanf("%d", &n);

    struct Job jobs[n];

    for (int i = 0; i < n; i++) {

        printf("Enter deadline and profit for job %d: ", i + 1);

        scanf("%d %d", &jobs[i].deadline, &jobs[i].profit);

        jobs[i].id = i + 1;

    }

     jobScheduling(jobs, n);

    return 0;

}

