课后作业 1

**1.** 环状数组区间最大和 1

**2.** 排序数组的平方 1

**3.** 包含01的N位数 1

**4.** Introvert让梨 2

**5.** (USACO 2014 December – Bronze 3) Cow Jog 3

**6.** (USACO 2018 February – Silver 1) Rest Stops 4

课后作业

1. 环状数组区间最大和

给定（）个整数（）。依次排列在环上，使得与相邻。给定，在环上取连续的（）个数，求取出的数之和的最大值。

输入样例：

|  |
| --- |
| 6 3  8 3 5 4 9 2 |

输出样例：

|  |
| --- |
| 19 |

1. 排序数组的平方

给定一个排序数组。构造一个数组，为中每个元素的平方排序后的结果。

|  |
| --- |
| vector<long long> sortedSquares(const vector<int> &a) |

输入样例：

|  |
| --- |
| 5  -3 -2 0 2 4 |

输出样例：

|  |
| --- |
| 0 4 4 9 16 |

本题的限制为：你的函数需要有的运行时间。

1. 包含01的N位数

给定一个整数（）。求满足以下条件的正整数的数量，结果对取模：

1. 包含且仅包含数位和数位的位数。
2. 包含数位且包含数位的位数。

输入样例：

|  |
| --- |
| 2 |

输出样例：

|  |
| --- |
| 1  1 |

输入样例：

|  |
| --- |
| 3 |

输出样例：

|  |
| --- |
| 3  35 |

1. Introvert让梨

孔融小朋友家里来了 N-1 个客人，他决定把桌子上 M 个一摸一样的梨分给客人和自己。现在知道对于在场的第 i 个人，如果拿到超过 K\_i 个梨子，TA 就会感到羞愧难当。给定 N, M, 和 {K\_i}，请计算无人羞愧难当的分梨方案数。

输入样例：

|  |
| --- |
| 4 8  3 2 2 2 |

输出样例：

|  |
| --- |
| 4 |

说明：

方案如下：

(3, 1, 2, 2), (3, 2, 1, 2), (3, 2, 2, 1), (2, 2, 2, 2).

1. (USACO 2014 December – Bronze 3) Cow Jog

[Mark Gordon, 2014]

The cows are out exercising their hooves again! There are N cows jogging on an infinitely-long single-lane track (1 <= N <= 100,000). Each cow starts at a distinct position on the track, and some cows jog at different speeds.

With only one lane in the track, cows cannot pass each other. When a faster cow catches up to another cow, she has to slow down to avoid running into the other cow, becoming part of the same running group.

Eventually, no more cows will run into each other. Farmer John wonders how many groups will be left when this happens. Please help him compute this number.

**INPUT: (file cowjog.in)**

The first line of input contains the integer N.

The following N lines each contain the initial position and speed of a single cow. Position is a nonnegative integer and speed is a positive integer; both numbers are at most 1 billion. All cows start at distinct positions, and these will be given in increasing order in the input.

**SAMPLE INPUT:**

|  |
| --- |
| 5  0 1  1 2  2 3  3 2  6 1 |

**OUTPUT: (file cowjog.out)**

A single integer indicating how many groups remain.

**SAMPLE OUTPUT:**

|  |
| --- |
| 2 |

1. (USACO 2018 February – Silver 1) Rest Stops

Farmer John and his personal trainer Bessie are hiking up Mount Vancowver. For their purposes (and yours), the mountain can be represented as a long straight trail of length meters (). Farmer John will hike the trail at a constant travel rate of seconds per meter (). Since he is working on his stamina, he will not take any rest stops along the way.

Bessie, however, is allowed to take rest stops, where she might find some tasty grass. Of course, she cannot stop just anywhere! There are rest stops along the trail (); the -th stop is meters from the start of the trail () and has a tastiness value (). If Bessie rests at stop for seconds, she receives tastiness units.

When not at a rest stop, Bessie will be hiking at a fixed travel rate of seconds per meter (). Since Bessie is young and fit, is strictly less than .

Bessie would like to maximize her consumption of tasty grass. But she is worried about Farmer John; she thinks that if at any point along the hike she is behind Farmer John on the trail, he might lose all motivation to continue!

Help Bessie find the maximum total tastiness units she can obtain while making sure that Farmer John completes the hike.

**INPUT FORMAT (file reststops.in):**

The first line of input contains four integers: , , , and . The next lines describe the rest stops. For each between and , the -st line contains two integers and , describing the position of the -th rest stop and the tastiness of the grass there.

It is guaranteed that , and . Note that and are given in seconds per meter!

**OUTPUT FORMAT (file reststops.out):**

A single integer: the maximum total tastiness units Bessie can obtain.

**SAMPLE INPUT:**

|  |
| --- |
| 10 2 4 3  7 2  8 1 |

**SAMPLE OUTPUT:**

|  |
| --- |
| 15 |

In this example, it is optimal for Bessie to stop for seconds at the rest stop (acquiring tastiness units) and then stop for an additional second at the rest stop (acquiring more tastiness unit, for a total of tastiness units).

Problem credits: Dhruv Rohatgi