The subway in town X is a bit unusual. One train consists of a single wagon and there is a single row of *L* seats in it. If there are *N* passengers in the wagon, numbered from 0 to *N-1*, each passenger gets a certain amount of pleasure as follows:

- If a passenger is standing, he gets pleasure equal to 0;
- Otherwise the passenger gets *A[i]* pleasure from sitting and additional *B[i]* pleasure for each empty seat between him and a neighboring passenger or between him and the end of the seats row.

For example let's have 3 passengers in the wagon, numbered 0, 1 and 2, and *A[0]*=5, *B[0]*=2, *A[1]*=10, *B[1]*=1, *A[2]*=1, *B[2]*=1. Let the number of seats *L*=6 and consider the passengers sitting in the following schema: _ 0 _ _ 1 _ ("_" means an empty seat)

Passenger 2 is standing.

In this case:

- Passenger 0 gets pleasure 5, because he is sitting + pleasure 6, because of the empty seats (1 to the left and 2 to the right). That's a total pleasure of 11.
- Passenger 1 gets pleasure 10, because he is sitting + pleasure 3, because of the empty seats (2 to the left and 1 to the right). Total of 13.
- Passenger 2 gets pleasure 0, because he is standing.

The total pleasure of all the passengers is equal to 24.

Write program **seats**, which, given the number of seats ***L***, the number of passengers ***N*** and the pleasure characteristics for each passenger, determines the maximum possible total pleasure for any number of seated passengers between 1 and ***N***.

Input

The first line of the standard input contains two integers – *N* and *L* - the number of passengers and the number of seats.

Each of the next *N* lines contains two non-negative integers – the characteristics *A[i]* and *B[i]* for passenger with index *i*.

Output

Print *N* lines, the *K*-th of which contains a single number – the maximum total pleasure that the passengers can get if there are exactly *K* of them sitting.

(For $K > L$ print 0, because there are no valid configurations of *K* passengers on *L* seats)

Constraints

$1 \le N \le 100\,000$
$1 \le \boldsymbol{L} \le 200\,000$

$0 < A[i], B[i] < 10^9$

Subtasks

Subtask 1 (20 points): $1 \le \boldsymbol{N} \le 200$
Subtask 2 (30 points): $1 \le \boldsymbol{N} \le 5000$

Subtask 3 (50 points): There are no additional constraints for the other test cases

To get the points for a given subtask your program should pass all the test cases of the subtask.

Example 1

Input	*Output*
3 2 1 2 3 4 5 6	11 8 0

Example 2

Input	*Output*
3 3 1 2 3 4 5 100	205 112 9

Example 1: For $K = 2$ the optimal configuration is: 1,2. Then passenger 1 gets pleasure 3, for being seated and 0*4, because there are no empty seats between him and the other passengers. Similarly, passenger 2 gets pleasure 5+0*6. Passenger 0 is standing so he gets pleasure 0. In total: 0 + (3+0*4) + (5+0*6) = 8.

Example 2: For $K = 1$ the optimal configuration is: 2,_,_. The total pleasure is: 0 + 0 + (5+2*100) = 205.