

**Task – homecoming**

Time limit:        **0.3 seconds**  
 Memory limit:    **256 megabytes**

Spiderman (Peter) is still in high school. Our friendly neighborhood superhero has  $N$  subjects at school numbered from  $0$  to  $N-1$ . For each subject Peter passes, he receives a prize in money from Tony Stark. If Peter passes subject number  $i$ , he receives  $A_i$  dollars.

Passing a subject is unfortunately not that easy. In order to pass he needs to buy some textbooks. Of course, our friendly superhero is very smart, so he doesn't need any textbook to study, but some teachers just won't let him pass unless he invests some money in the books. There are  $N$  textbooks numbered from  $0$  to  $N - 1$ , the  $i$ -th of which costs  $B_i$  dollars.

In order to pass subject number  $i$ , Peter needs to buy textbooks  $i$ ,  $(i + 1) \% N$ ,  $(i + 2) \% N$ , ...,  $(i + K - 1) \% N$ , where  $K$  is a given constant.

Peter doesn't care about school anymore since his dream is to become an Avenger, so it's not relevant whether he passes all subjects or not. Peter loves time, and time is money, so help Peter make the biggest profit.

**Interaction**

The contestant should include the header `homecoming.h` (C / C++) or implement `homecoming.pas` (Pascal).

The contestant needs to implement the following function:

```
long long int solve(int N, int K, int *A, int *B);
function solve(N, K: LongInt, var A, B: array of LongInt): Int64;
```

This function may be called multiple times over the course of a single run.

**Constraints**

Let  $S_N$  be the sum of all  $N$ 's over all calls of `solve`, and let  $S_{NK}$  be the sum of  $N \cdot K$  over all calls of `solve`. Then:

- $1 \leq K \leq N \leq 2.000.000$
- $1 \leq S_N \leq 2.000.000$
- $0 \leq A_i, B_i \leq 1.000.000.000$

**Scoring****Subtask 1** (13 points)

- $1 \leq S_N \leq 500$

**Subtask 2** (18 points)

- $1 \leq S_N \leq 5.000$

**Task – homecoming****Subtask 3** (31 points)

- $1 \leq S_{NK} \leq 2.000.000$

**Subtask 4** (38 points)

- No additional constraints.

**Example**

call	returned value
<code>solve(3, 2, [40, 80, 100], [140, 0, 20])</code>	60

**Notes**

- Given two positive numbers A and B,  $A \% B$  denotes the remainder of A when divided by B.
- The visible tests will not be grouped with other tests.