7773 Delight for a Cat

A cat is going on an adventure.

Each hour, the cat can be either *sleeping* or *eating*. The cat cannot be doing both actions at the same hour, and the cat is doing exactly one of these actions for the whole hour.

For each of the next n hours, the amount of *delight* the cat is getting if it is sleeping or eating during that hour is known. These amounts can be different for each hour.

An integer time period k is also known. Among every k consecutive hours, there should be at least m_s hours when the cat is sleeping, and at least m_e hours when the cat is eating. So, there are exactly $n - k + 1$ segments of k hours for which this condition must be satisfied.

Find the maximum total amount of delight the cat can get during the next n hours.

Input

The input file contains several test cases, each of them as described below.

The first line of the input contains four integers n, k, m_s, and m_e ($1 \le k \le n \le 1000$; $0 \le ms, me \le k$; $m_s + m_e \le k$) — the number of upcoming hours, the length of the period (in hours), and the minimum number of hours the cat should be sleeping and eating out of every k consecutive hours, respectively.

The second line contains n integers $s_1, s_2, \ldots, s_n$ ($0 \le s_i \le 10^9$) — the amount of delight the cat gets when it is sleeping during the first, the second, ..., the n-th hour.

The third line contains n integers $e_1, e_2, \ldots, e_n$ ($0 \le e_i \le 10^9$) — the amount of delight the cat gets when it is eating during the first, the second, ..., the n-th hour.

Output

For each test case, the output must follow the description below.

In the first line, output a single integer — the maximum total amount of delight the cat can get during the next n hours.

In the second line, output a string of length n consisting of characters '`S`' and '`E`'. The i-th character of this string should correspond to whether the cat should sleep (`S`) or eat (`E`) in the i-th hour to get the maximum total amount of delight out of these n hours.

Sample Input

```
10 4 1 2
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
```

Sample Output

```
69
EEESESEESS
```