

CCO '11 P5 - Fixing Disks

Time limit: 1.0s **Memory limit:** 1G

Canadian Computing Competition: 2011 Stage 2, Day 2, Problem 2

You are playing a game with a stack of N disks ($1 \leq N \leq 100$). The goal of the game is to remove all of the disks from your stack. However, there is a cost associated with removing disks, and you wish to minimize the cost of removing all the disks from your stack.

Each disk contains a label, with the label L being in the range $1 \leq L \leq 20$.

You are also given a "Master stack" of N disks which you can use to help you remove disks from your stack.

You can remove disks from the top of your stack of disks in two ways:

1. remove a disk at the top of your stack: if label of the disk on top of your stack is c , that disk can be removed from your stack with cost c ;
2. remove a disk from both the top of the Master stack and the top of your stack if the label is the same between both disks: in this case, there is no cost with removing both disks.

You are also allowed to modify the order of the top K ($1 \leq K \leq 4$) elements of your stack, so long as immediately after each modification, you remove the top element of your stack. There are three allowed modifications:

1. Reverse: you may reverse the order of the top r ($2 \leq r \leq K$) disks on your stack. In other words, if the top r disks are $d_1, d_2, \dots, d_r$ (reading from the top down), then after this operation, the top r disks will be $d_r, \dots, d_2, d_1$ (reading from the top down). The cost of one reverse operation is R ($1 \leq R \leq 1\,000\,000$).
2. (Cyclic Shift) Up: you can shift up one disk in the range of the top u disks ($2 \leq u \leq K$). For example, if the top four disks are d_1, d_2, d_3, d_4 (read from the top down), you can perform an up shift in the range of the top three elements to get d_2, d_3, d_1, d_4 or an up shift of the top four elements to get d_2, d_3, d_4, d_1 . The cost of one up shift operation is U ($1 \leq U \leq 1\,000\,000$).
3. (Cyclic Shift) Down: you can shift down one disk in the range of the top d disks ($2 \leq d \leq K$). For example, if the top four disks are d_1, d_2, d_3, d_4 (read from the top down), you can perform a down shift in the range of the top three elements to get d_3, d_1, d_2, d_4 or a down shift of the top four elements to get d_4, d_1, d_2, d_3 . The cost of one down shift operation is D ($1 \leq D \leq 1\,000\,000$).

If the operations yield a match between the top of the master stack and the top of your stack, you can pop for free. If not, you must pay the cost of the pop.

There is one additional constraint. At any point in the game, if a disk at level j is being popped (levels start at 0 at the bottom of the stack), then all elements that were *originally* at level $j + M$ or higher must have already been popped, where $1 \leq M \leq 5$.

Minimize the cost required to pop all the elements off of your stack.

Input Specification

The first line will contain 6 space-separated integers: N, K, M, D, U, R where:

- N is the number of disks in each of the stacks ($1 \leq N \leq 100$)
- K , the maximum depth at which operations can be done ($1 \leq K \leq 4$)
- M , the threshold before which elements can be removed from our stack, ($1 \leq M \leq 5$)
- D , the cost for a shift in which the bottom element of the selected range goes to the top ($1 \leq D \leq 10^6$)
- U , the cost for an up shift operation (in which the top element of the selected range goes to the bottom), ($1 \leq U \leq 10^6$)
- R , the cost for reversing the top elements of the selected range ($1 \leq R \leq 10^6$)

$2N$ lines will follow, each containing a number L ($1 \leq L \leq 20$). The first N lines will contain the labels of the disks in the Master stack, from top to bottom. The next N lines will contain the labels of the disks in your stack, from top to bottom.

Note: For 20% of the marks for this question, you may assume that $K \leq 2$.

Output Specification

Output the integer (on one line) which is the minimal cost required to remove all disks from your stack.

Sample Input

```
7 3 3 4 4 3
5
6
3
5
4
1
2
3
5
6
5
1
4
1
```

Sample Output

```
5
```

Explanation for Sample Output

We take 3 to the bottom and shift the two blocks below it up, with cost 4. Then we remove four blocks from each stack, remove a 1 from the playing stack, with cost 1, and remove two blocks from each stack.