Indian National Olympiad in Informatics, 2009

Time: 3 hours   *24 January, 2009*

Instructions

(a) You will have to return this question paper at the end of the examination with relevant parts filled out.

(b) There are two questions. You have to write working programs in Pascal, C or C++ to solve each of these questions.

- *Only your source code will be submitted for evaluation. Your program will be recompiled and run on the evaluation computer.*

- *Make sure your C/C++ programs compile with the GNU C compiler (**djgpp**, under Windows). Programs written using Turbo C++ may not compile and run in the evaluation environment and may hence lose marks.*

(c) At the end of each question, there is a space to indicate the location of the source code file for your solution. Please fill up this information without fail. Otherwise, your solution cannot be evaluated.

(d) All input for your programs will come from the keyboard. All output from your programs should be written to the screen.

(e) Please fill out your contact details on the reverse of this page as completely as you can. Use your serial number as reported in the ZIO/ZCO results page as your roll number for INOI. If you qualified under multiple categories, use the ZIO serial number as your roll number. Ask your centre supervisor if you do not know your roll number.

(f) At the beginning of each program, include a comment with your name, roll number and the problem number. For instance, if your name is Sachin Tendulkar, your roll number is 107 and the problem you are writing a C/C++ program for Problem 1, add a comment of the form

```
/*  Sachin Tendulkar, 107, Problem 1 */
```

at the top of your file.

This will help us if files accidentally get mixed up when being sent from the exam centres to the evaluation centre.

Contact details	
Roll Number:	
Name:	
Home Address: PIN Code	
Phone Number(s): (with STD code)	
Email address:	

Question 1   Rearranging cartons

Your school has ordered some equipment that has been delivered in a number of very heavy cartons. Each carton has a serial number and the cartons are all lined up in a row. Unfortunately, your teacher asked for the cartons to be placed in a particular sequence and you forgot to tell the people who unloaded the cartons about this. You now have to quickly restore the cartons to the correct order before the teacher comes and sees how you have messed up her instructions.

Since the cartons are very heavy, you cannot carry them over long distances. In each step, all you can do is to exchange the position of two adjacent cartons.

For instance, suppose the serial numbers on the cartons in the order in which they are unloaded are 34, 29, 12, 78 and 90 and the order in which the cartons were supposed to be arranged is 90, 29, 78, 34, 12. These cartons can be rearranged in the desired order with 7 exchanges, as follows:

- Exchange 78, 90 — 34, 29, 12, <u>90, 78</u>

- Exchange 12, 90 — 34, 29, <u>90, 12</u>, 78

- Exchange 34, 29 — <u>29, 34</u>, 12, 90, 78

- Exchange 12, 78 — 29, 34, 90, <u>78, 12</u>

- Exchange 34, 90 — 29, <u>90, 34</u>, 78, 12

- Exchange 29, 90 — <u>90, 29</u>, 34, 78, 12

- Exchange 34, 78 — 90, 29, <u>78, 34</u>, 12

In this example, it can be shown that 7 exchanges are needed to reorder the cartons as desired.

Clearly, you want to get the job done with minimum effort. Given the initial arrangement of the cartons and the final sequence that the teacher wants, your goal is to compute the minimum number of exchanges required to rearrange the cartons in the desired order.

Input format

The first line of input is a single integer N, the total number of cartons. The second line consists of N distinct positive integers, separated by spaces, denoting the serial numbers of the N cartons in the order in which they were unloaded. The third line is another sequence of N integers, denoting the desired order in which the N cartons should be rearranged. The sequence of numbers in the third line is guaranteed to be a permutation of the sequence in the second line.

Output format

The output should be a single integer, the minimum number of exchanges required to achieve the desired sequence of cartons.

Test data

In 90% of the inputs, $1 \leq N \leq 6000$. In 10% of the inputs, $1 \leq N \leq 100000$.

Example

Here is the sample input and output corresponding to the example discussed above.

Sample input

```
5
34 29 12 78 90
90 29 78 34 12
```

Sample output

```
7
```

Note: Your program should not print anything other than what is specified in the output format. Please remove all diagnostic print statements before making your final submission. A program with extraneous output will be treated as incorrect!

Important

Indicate the location of the source code file for your solution to Question 1 in the box below. *If you fail to do this, your solution cannot be evaluated!*

Source file:

Question 2   Glass Bead Game

You play the following game against a devilishly clever opponent. On the table, there are two bowls filled with glass beads. In each move, depending on whose turn it is, you or your opponent can do one of the following.

- Move one bead from the first bowl to the second bowl.

- Remove one bead from the second bowl.

At any point in the game, let x denote the number of beads in the first bowl and y denote the number of beads in the second bowl. If $x + y$ is even, it is your turn to move. If $x + y$ is odd, it is your opponent's turn to move.

At each position of the game, you get some points that depend on the number of beads, x and y, in the two bowls. This is given by a function $f(x, y) = Ax^2 + By^2 + Cxy$, where A, B and C are integers that are fixed each time the game is played.

The game starts with M beads in the first bowl and N beads in the second bowl. As the game progresses, beads are removed. The game ends when both bowls are empty.

Your score for the game is the sum of the values $f(x, y)$ for each position (x, y) that the game passes through, starting from (M, N) and ending with $(0, 0)$. Your aim is to obtain as high a score as possible, no matter what moves your opponent chooses. Remember that your opponent is fiendishly ingenious and will always play in such a way as to minimize your score.

For example, suppose the game starts with 2 beads in each bowl and the score function is $f(x, y) = x^2 - y^2 - xy$: that is, $A = 1$, $B = -1$ and $C = -1$. Here are some plays of the game—the positions where you choose the move are enclosed in boxes and the positions where your opponent moves are not enclosed in boxes.

- $\boxed{(2,2)} \rightarrow (2,1) \rightarrow \boxed{(2,0)} \rightarrow \boxed{(1,1)} \rightarrow (1,0) \rightarrow (0,1) \rightarrow (0,0)$.

  For this sequence of moves, your total score is $f(2, 2) + f(2, 1) + f(2, 0) + f(1, 1) + f(1, 0) + f(0, 1) + f(0, 0) = (-4) + 1 + 4 + (-1) + 1 + (-1) + 0 = 0$.

- $\boxed{(2,2)} \rightarrow \boxed{(1,3)} \rightarrow \boxed{(0,4)} \rightarrow (0,3) \rightarrow \boxed{(0,2)} \rightarrow (0,1) \rightarrow (0,0)$

  For this sequence of moves, your total score is $f(2, 2) + f(1, 3) + f(0, 4) + f(0, 3) + f(0, 2) + f(0, 1) + f(0, 0) = (-4) + (-11) + (-16) + (-9) + (-4) + (-1) + 0 = -45$.

In this particular game, the best possible score that you can achieve if you consider all possible moves that your opponent can choose is -22. Recall that your devilishly clever opponent always plays in such a way that your score is minimized.

Input format

The input consists of a single line containing five integers, M, N, A, B and C where M and N are the number of beads initially in the first and second bowl, respectively and A, B and C are the coefficients of the score function $f(x, y) = Ax^2 + By^2 + Cxy$.

Output format

The output should be a single integer, the maximum score that you can achieve in this game.

Test data

In all inputs, $1 \leq M, N \leq 600$ and $-1000 \leq A, B, C \leq 1000$.

Example

Sample input

```
2 2 1 -1 -1
```

Sample output

```
-22
```

Note: Your program should not print anything other than what is specified in the output format. Please remove all diagnostic print statements before making your final submission. A program with extraneous output will be treated as incorrect!

Important

Indicate the location of the source code file for your solution to Question 2 in the box below. *If you fail to do this, your solution cannot be evaluated!*

Source file: