ACM Algorithm Practice **Spring 2015**

Week 3 Problems

April 14, 2015 *Timothy Johnson*

Problem 1: Error Correct System

(Taken from http://codeforces.com/problemset/problem/527/B)

Ford Prefect got a job as a web developer for a small company that makes towels. His current work task is to create a search engine for the website of the company. During the development process, he needs to write a subroutinge for comparing strings S and T of equal length to be "similar". After a brief search on the Internet, he learned about the *Hamming distance* between two strings S and G of the same length, which is defined as the number of positions in which S and T have two different characters. For example, the Hamming distance between the words "permanent" and "pergament" is two, as these words differ in the fourth and sixth letters.

Moreover, as he was searching for information, he also noticed that modern search engines have powerful mechanisms to correct errors in the request to improve the quality of the search. Ford doesn't know much about human beings, so he assumed that the most common mistake in a request is swapping two arbitrary letters of the string (not necessarily adjacent). Now he wants to write a function that determines which two letters should be swapped in string S, so that the Hamming distance between a new string S and string T would be as small as possible, or otherwise, determines that such a replacement cannot reduce the distance between the strings.

Help him do this!

Input:
The first line contains the integer n ($1 \leq n \leq 200\ 000$), the length of strings S and T.

The second line contains the string S.

The third line contains the string T.

Each of these lines contains only lowercase Latin letters (a - z).

Output:
On the first line, print the minimum possible Hamming distance between strings S and T if you swap at most one pair of letters in S.

On the second line, either print the indices i and j ($1 \leq i, j \leq n,\ i \neq j$), if reaching the minimum possible distance is possible by swapping letters on positions i and j, or print "$-1\ \ -1$" if it is not necessary to swap characters.

If there are multiple possible answers, print any of them.

Examples:

- **Input:**
  9
  pergament
  permanent

  Output:
  1
  4 6

- **Input:**
  6
  wookie
  cookie

  Output:
  1
  -1 -1

- **Input:**
  4
  petr
  egor

  Output:
  2
  1 2

Problem 2: Group Photo 2

(Taken from http://codeforces.com/problemset/problem/529/B)

Many years have passed, and n friends met at a party again. Technologies have leaped forward since the last meeting, cameras with timers appeared and now it is not obligatory for one of the friends to stand with a camera, and, thus, being absent on the photo.

Simply speaking, the process of photographing them can be described as follows. Each friend occupies a rectangle of pixels on the photo. The i-th of them in a standing state occupies a rectangle that is w_i pixels wide and h_i pixels high. But also, each person can lie down for the photo, and then he will occupy a rectangle that is h_i pixels wide and w_i pixels high.

The total photo will have size $W \times H$, where W is the total width of all the people's rectangles, and H is the maximum of the heights. The friends want to determine the minimum area the group photo if no more than $n/2$ of them lie on the ground. Help them to achieve this goal.

Input:
The first line contains an integer n ($1 \leq n \leq 1000$), the number of friends.

The next n lines have two integers w_i, h_i ($1 \leq w_i, h_i \leq 1000$) each, representing the size of the rectangle corresponding to the i-th friend.

Output:
Print a single integer equal to the minimum possible area of the photo containing all the friends if no more than $n/2$ of them can lie on the ground.

Examples:

- **Input:**
  3
  10 1
  20 2
  30 3

  Output:
  180

- **Input:**
  3
  3 1
  2 2
  4 3

  Output:
  21

Problem 3: Fox and Jumping

(Taken from http://codeforces.com/problemset/problem/510/D)

Fox Ciel is playing a game. In this game there is an infinite long tape with cells indexed by integers (positive, negative and zero). At the beginning she is standing at the cell 0.

There are also n cards, each card has 2 attributes: length l_i and cost c_i. If she pays c_i collars then she can apply the i-th card. After applyting the i-th card she becomes able to make jumps of length l_i, i.e., from cell x to either cell $(x - l_i)$ or cell $(x + l_i)$.

She wants to be able to jump to any cell on the tape (possibly, visiting some intermediate cells). For achieving this goal, she wants to buy some cards, paying as little money as possible.

If this is possible, calculate the minimal cost.

Input:
The first line contains an integer n $(1 \le n \le 300)$, the number of cards.

The second line contains n number l_i $(1 \le l_i \le 10^9)$, the jump lengths of cards.

The third line contains n numbers c_i $(1 \le c_i \le 10^5)$, the costs of cards.

Output:
If it is impossible to buy some cards and become able to jump to any cell, output -1. Otherwise output the minimal cost of buying such set of cards.

Examples:

- **Input:**
  3
  100 99 9900
  1 1 1

  Output:
  2

- **Input:**
  5
  10 20 30 40 50
  1 1 1 1 1

  Output:
  -1

- **Input:**
  7
  15015 10010 6006 4290 2730 2310 1
  1 1 1 1 1 1 10

Output:
6

- **Input:**
  8
  4264 4921 6321 6984 2316 6120 1026
  4264 4921 6321 6984 2316 6120 1026

  Output:
  7237