1. Almost Equivalent Strings

Two strings are considered “almost equivalent” if they have the same length AND for each lowercase letter x, the number of occurrences of x in the two strings differs by no more than 3. There are two arrays of n strings, arrays s and t. Strings s[i] and t[i] make the ith pair. They are of equal length and consist of lowercase English letters. For each pair of strings, determine if they are almost equivalent. Return an array of i strings, either 'YES' or 'NO', one for each pair.

Example

s = ['aabaab', 'aaaaabb']

t = ['bbabbc', 'abb']

	s[0]	t[0]	difference
a	4	1	3
b	2	4	2
c	0	1	1

	s[1]	t[1]	difference
a	5	1	4
b	2	2	0

The number of occurrences of 'a', 'b', and 'c' in (s[0], t[0]) never differs by more than 3. This pair is almost equivalent.

In (s[1], t[1]), 'a' violates the condition.

The return array is ['YES', 'NO'].

Function Description

Complete the function areAlmostEquivalent in the editor below.

areAlmostEquivalent has the following parameters:

string s[n]: an array of strings

string t[n]: an array of strings

Returns:

string[n]: an array of strings, either 'YES' or 'NO' in answer to each test case

Constraints

1 ≤ n ≤ 5
1 ≤ length of any string in the input ≤ 105

Input Format For Custom Testing
Sample Case 0
Sample Input

STDIN   Function
-----   --------
1   → s[] size n = 1
aaa   → s = ['aaa']
1   → t[] size n = 1
aab   → t = ['aab']
Sample Output

YES
Explanation

Only letters 'a' and 'b' are present in the two strings. The number of occurrences of each differs by 1.

2. Cutting Metal Surplus

The owner of a construction company has a surplus of rods of arbitrary lengths. A local contractor offers to buy any of the surplus, as long as all the rods have the same exact integer length, referred to as saleLength. The number of sellable rods can be increased by cutting each rod zero or more times, but each cut has a cost denoted by costPerCut. After all cuts have been made, any leftover rods having a length other than saleLength must be discarded for no profit. The owner's total profit for the sale is calculated as:

totalProfit = totalUniformRods × saleLength × salePrice − totalCuts × costPerCut

where totalUniformRods is the number of sellable rods, salePrice is the per unit length price that the contractor agrees to pay, and totalCuts is the total number of times the rods needed to be cut.

Example

lengths = [30, 59, 110]

costPerCut = 1

salePrice = 10 per unit length

The following are tests based on lengths that are factors of 30, the length of the shortest bar. Factors of other lengths might also be tested, but this demonstrates the methodology.

Cuts

```
                      ---------------------
Length                Rod     Extra   Regular Total   Pieces
----------------------------------------------
30        30          0       0       0       1
          59          1       0       1       1
          110         1       2       3       3
          Revenue = (10*5*30)-(4*1) = 1496
15        30          0       1       1       2
          59          1       2       3       3
          110         1       6       7       7
          Revenue = (10*12*15)-(11*1) = 1789
10        30          0       2       2       3
          59          1       4       5       5
          110         0       10      10      11
          Revenue = (10*19*10)-(17*1) = 1883
6         30          0       4       4       5
          59          1       8       9       9
          110         1       17      18      18
          Revenue = (10*32*6)-(31*1) = 1889
5         30          0       5       5       6
          59          1       10      11      11
          110         0       21      21      21
          Revenue = (10*39*5)-(37*1) = 1913
3         30          0       9       9       10
          59          1       18      19      19
          110         1       35      36      36
          Revenue = (10*65*3)-(64*1) = 1886
```

Working through the first stanza, length = 30, it is the same length as the first rod, so no cuts are required and there is 1 piece. For the second rod, cut and discard the excess 29 unit rod. No more cuts are necessary and another 1 piece is left to sell. Cut 20 units off the 110 unit rod to discard leaving 90 units, then make two more cuts to have 3 more pieces to sell. Finally sell 5 totalUniformRods , saleLength = 30 at salePrice = 10 per unit length for 1500. The cost to produce was totalCuts = 4 times costPerCut = 1 per cut, or 4. Total revenue = 1500-4=1496. The maximum revenue among these tests is obtained at length 5 for 1913.

Function Description

Complete the function maxProfit in the editor below.

maxProfit has the following parameter(s):

costPerCut: cost to make a cut

salePrice: per unit length sales price

lengths[n]: integer rod lengths

Returns:

int: maximum possible profit

Constraints

1 ≤ n ≤ 50
1 ≤ lengths[i] ≤ 104
1 ≤ salePrice, costPerCut ≤ 1000

Input Format for Custom Testing
Sample Case 0
Sample Input

```
STDIN     Function
-----     -----
1      →  costPerCut = 1
10     →  salePrice = 10
3      →  lengths[] size n = 3
26     →  lengths = [26, 103, 59]
103
59
```

Sample Output

1770

Explanation

Since costPerCut = 1 is very inexpensive, a large number of cuts can be made to reduce the number of wasted pieces. The optimal rod length for maximizing profit is 6, and the rods are cut as shown:

lengths[0] = 26: Cut off a piece of length 2 and discard it, resulting in a rod of length 24.
Then, cut this rod into 4 pieces of length 6.
lengths[1] = 103: Cut off a piece of length 1 and discard it, resulting in a rod of length 102.
Then, cut this rod into 17 pieces of length 6.
lengths[2] = 59: Cut off a piece of length 5 and discard it, resulting in a rod of length 54.
Then, cut this rod into 9 pieces of length 6.

After performing totalCuts = (1 + 3) + (1 + 16) + (1 + 8) = 30 cuts, there are totalUniformRods = 4 + 17 + 9 = 30 pieces of length saleLength = 6 that can be sold at salePrice = 10. This yields a total profit of salePrice × totalUniformRods × saleLength − totalCuts × costPerCut = 10 × 30 × 6 − 30 × 1 = 1770.

Sample Case 1