ELectronics Engineering Students' Association (ELESA)
Presents

ELESA Placement Cell (EPC)

Assessment test

Name of the Candidate: Mail ID:	Section (No of Questions): Coding Duration:

Diagonal Difference

Given a square matrix of size $N \times N$, calculate the absolute difference between the sums of its diagonals.

Input Format

The first line contains a single integer, N. The next N lines denote the matrix's rows, with each line containing N space-separated integers describing the columns.

Constraints

- $-100 \leq \text{Elements in the matrix} \leq 100$

Output Format

Print the absolute difference between the two sums of the matrix's diagonals as a single integer.

Sample Input

```
3
11 2 4
 4 5 6
10 8 -12
```

Sample Output

```
15
```

Explanation

The primary diagonal is:

```
11
  5
    -12
```

Sum across the primary diagonal: 11 + 5 - 12 = 4

The secondary diagonal is:

```
   4
  5
 10
```

Sum across the secondary diagonal: 4 + 5 + 10 = 19
Difference: |4 - 19| = 15

Note: |x| is absolute value function

Making Anagrams

We consider two strings to be anagrams of each other if the first string's letters can be rearranged to form the second string. In other words, both strings must contain the same exact letters in the same exact frequency. For example, bacdc and dcbac are anagrams, but bacdc and dcbad are not.

Alice is taking a cryptography class and finding *anagrams* to be very useful. She decides on an encryption scheme involving two large strings where encryption is dependent on the minimum number of character deletions required to make the two strings anagrams. Can you help her find this number?

Given two strings, $s1$ and $s2$, that may not be of the same length, determine the minimum number of character deletions required to make $s1$ and $s2$ anagrams. Any characters can be deleted from either of the strings.

For example, $s1$=**abc** and $s2$=**amnop**. The only characters that match are the **a**'s so we have to remove **bc** from $s1$ and **mnop** from $s2$ for a total of 6 deletions.

Function Description

Complete the *makingAnagrams* function in the editor below. It should return an integer representing the minimum number of deletions needed to make the strings anagrams.

makingAnagrams has the following parameter(s):

- *s1*: a string
- *s2*: a string

Input Format

The first line contains a single string, $s1$.
The second line contains a single string, $s2$.

Constraints

- $1 \leq |s1|, |s2| \leq 10^4$
- It is guaranteed that $s11$ and $s22$ consist of lowercase English letters, ascii[a-z].

Output Format
Print a single integer denoting the minimum number of characters which must be deleted to make the two strings anagrams of each other.

Sample Input

```
cde
abc
```

Sample Output

```
4
```

Explanation
We delete the following characters from our two strings to turn them into anagrams of each other:
1. Remove d and e from cde to get c .
2. Remove a and b from abc to get c
We had to delete 4 characters to make both strings anagram.

Printing Pattern using Loops

In this problem, you need to print the pattern of the following form containing the numbers from 1 to n.

```
4 4 4 4 4 4 4
4 3 3 3 3 3 4
4 3 2 2 2 3 4
4 3 2 1 2 3 4
4 3 2 2 2 3 4
4 3 3 3 3 3 4
4 4 4 4 4 4 4
```

Input Format

The input will contain a single integer n.

Constraints

$1 \leq n \leq 1000$

Output Format

Print the pattern mentioned in the problem statement.

Sample Input 0

```
2
```

Sample Output 0

```
2 2 2
2 1 2
2 2 2
```

Sample Input 1

```
5
```

Sample Output 1

```
5 5 5 5 5 5 5 5 5
5 4 4 4 4 4 4 4 5
5 4 3 3 3 3 3 4 5
5 4 3 2 2 2 3 4 5
5 4 3 2 1 2 3 4 5
5 4 3 2 2 2 3 4 5
5 4 3 3 3 3 3 4 5
5 4 4 4 4 4 4 4 5
5 5 5 5 5 5 5 5 5
```

Sample Input 2

```
7
```

Sample Output 2

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

Time Conversion

Given a time in 12-hour AM/PM format, convert it to military (24-hour) time.

Note: Midnight is 12:00:00AM on a 12-hour clock, and 00:00:00 on a 24-hour clock. Noon is 12:00:00PM on a 12-hour clock, and 12:00:00 on a 24-hour clock.

Input Format

A single string s containing a time in 12-hour clock format (i.e.: $\mathtt{hh}:\mathtt{mm}:\mathtt{ss}\mathtt{AM}$ or $\mathtt{hh}:\mathtt{mm}:\mathtt{ss}\ \mathtt{PM}$), where $01 \leq hh \leq 12$ and $00 \leq mm, ss \leq 59$.

Output Format

Convert and print the given time in 24-hour format, where $00 \leq hh \leq 23$.

Sample Input

```
07:05:45PM
```

Sample Output

```
19:05:45
```