Lab 2

Due Date: August 13.

A string in C can be defined as follows:

```
char name[10] = "DSA";
```

In the above case, name[0] = 'D', name[1]='S', name[2] ='A' and name[3]='\0'. The last character is called a null character and is added to the end of the string by default. This implies that it is better to store a string of size 9 in name.

The following assignment is not allowed in C

```
name = "DSA2";
```

However, we can change the individual elements of the string as follows:

```
name[0] = 'L';
```

This will change $name$ from DSA to LSA. We can take a string as an input from the user as follows:

```
scanf("%s", name);
```

First, note that, unlike int and float, we do not write &name but just name. Similarly, for printing a string, we use the following code.

```
printf("%s", name);
```

To find the length of the string, we can use the following function from string.h

```
#include<stdio.h>
#include<string.h>

int main()
{
  char name[10];
  scanf("%s", name);
  int len = strlen(name);
  printf("The length of name is %d", len);
}
```

1. String Reverse

   Given a string as an input, write a program that reverses the string and prints it.

   Input: A single string on a single line. The length of the string will be ≤ 1000.

   Output: The string written in reverse.

Input	:	DSA1
Output	:	1ASD

2. Telephone Number

   A telephone number is a sequence of 10 characters where the first character must be 9. Given a string, in an operation, you can delete any character from the string. For example, from the string 1234, you can make the following four strings $\{123, 124, 134, 234\}$. You can perform this operation as many times as you want.

   Given a string, your job is to find if there is a sequence of operations after which you are left with a valid telephone number.

Input: A single line that contains a positive number. The number of digits in this number will be ≤ 100.

Output: Yes, if the string can be converted to a valid phone number. No, otherwise.

Input	:	8788787900034
Output	:	No

Input	:	8999787900034
Output	:	Yes

3. k-string

A string is called a k-string if it can be represented as k concatenated copies of some string. For example, the string "aabaabaabaab" is at the same time a 1-string, a 2-string, and a 4-string, but it is not a 3-string, a 5-string, or a 6-string and so on. Obviously, any string is a 1-string.

You are given a string s, consisting of lowercase English letters and a positive integer k. Your task is to reorder the letters in the string s in such a way that the resulting string is a k-string.

Input: The first input line contains integer $k(1 \leq k \leq 000)$. The second line contains s, all characters in s are lowercase English letters. The string length s satisfies the inequality $1 \leq |s| \leq 1000$, where $|s|$ is the length of string s.

Output: Yes, if there is a way to arrange the string to get a k-string. No, otherwise

Input	:	2 abba
Output	:	Yes

Explanation: We can rearrange the string as $abab$ – this string is a 2-string.

Input	:	2 abbz
Output	:	No

4. Inversion

Given an array A, there is an inversion between index $A[i]$ and $A[j]$ in A if $i < j$ but $A[i] > A[j]$. Find the number of inversions in the array A.

Input: The first line of the input will be n – the number of elements in the array. This number will be ≤ 1000. The second line of the input will be n integers all of them will be between $[-1000, 1000]$.

Output: A single integer that is the number of inversions in the array.

Input	:	5 1 9 6 4 5
Output	:	5

Explaination: The following pairs are in inversions – $(9, 6)(9, 4)(9, 5)(6, 4)(6, 5)$

5. Permutation Sequence

You are given an array A. You can select any k numbers in this array and permute them in any way you want. This operation can be done just once. The rest of the numbers remain untouched. Our goal is to find a minimum k such that after permuting these k numbers, A is sorted.

Input: The first line of the input is n – the number of elements in the array ($n \leq 1000$). The second line is n number all of them between $[-1000, 1000]$.

Output: The number k.

Input	:	5 1 9 5 4 6
Output	:	3

Explanation: We select the numbers 9 4 6 and permute them to 4 6 9.

Input : 5
5 2 3 1 4
Output : 3

Explanation: We select numbers 5 1 4 and permute it to 1 4 5