Check Palindrome

Given a string, write a c function to check if it is palindrome or not.

A string is said to be palindrome if reverse of the string is same as string. For example, “abba” is palindrome, but “abbc” is not palindrome.

Input Format

In the function a string is passed.

Output Format

Return true if it is palindrome else false.

Sample Input

```
abcdcba
```

Sample Output

```
true
```

Solution: palindrome.cpp

String Compression

Given an array of characters `chars`, compress it using the following algorithm:

Begin with an empty string `s`. For each group of **consecutive repeating characters** in `chars`:

- If the group's length is 1, append the character to `s`.
- Otherwise, append the character followed by the group's length.

The compressed string `s` **should not be returned separately**, but instead be stored **in the input character array** `chars`. Note that group lengths that are 10 or longer will be split into multiple characters in `chars`.

After you are done **modifying the input array**, return *the new length of the array*.

You must write an algorithm that uses only constant extra space.

Input Format

In the function a vector of characters is passed.

Output Format

Return the updated vector

Example 1:

```
Input: chars = ["a","a","b","b","c","c","c"]Output: Return 6, and the first 6 characters of the input array should be:
["a","2","b","2","c","3"]Explanation: The groups are "aa", "bb", and "ccc". This compresses to "a2b2c3".
```

Example 2:

```
Input: chars = ["a","b","b","b","b","b","b","b","b","b","b","b","b"]Output: Return 4, and the first 4 characters of the input array should be:
["a","b","1","2"].Explanation: The groups are "a" and "bbbbbbbbbbbb". This compresses to "ab12".
```

Solution: stringCompression.cpp

Are Permutation

Given two strings A and B. Check if one string is permutation of the other.
A Permutation of a string is another string that contains same characters, only the order of characters can be different. For example, “abcd” and “dabc” are Permutation of each other.

Input Format

In the function two strings passed.

Output Format

Return true if B is permutation of A else false.

Sample Input

```
string A = "test", B = "ttew"
```

Sample Output

```
NO
```

Solution: premutation.cpp

Remove Duplicates

Given a string **S**, the task is to remove all the duplicates from the given string

and return the updated string in sorted order.

Input Format

In the function a string is passed.

Output Format

Return the updated string.

Sample Input

```
string s = "geeksforgeeks"
```

Sample Output

```
"efgkors"
```

Solution: removeDuplicates.cpp

Vowel Find

Given a string consisting of lowercase English alphabets, return a string containing all the vowels present in S in serial order.

Input Format

In the function a string S is passed.

Output Format

Return a string.

Sample Input

```
S = "aeoibsddaeioudb"
```

Sample Output

```
"aeoiaeiou"
```

Solution: vowels.cpp

Binary String to Number

\Given a binary string as input, convert into its decimal form and return it as an integer.

Input Format

In the function a binary string is passed.

Output Format

Return an integer.

Sample Input

```
111
```

Sample Output

```
7
```

Solution: binaryString.cpp

Search All!

Implement a function that returns a list of all occurrences of a given `substring` inside a `big string`.
Return empty vector if smaller string is not present inside bigger string.

Sample Input

```
string bigString = "I liked the movie, acting in movie was great!";string smallString = "movie"
```

Sample Output

```
12, 29
```

Solution: searchAll.cpp

Digital Clock

You are building a logic for a clock that requires you convert absolute time in minutes into a format supported by a digital clock. See examples below.

125 minutes can be displayed as **2:05**

155 minutes can be displayed as **2:35**

(You can assume the maximum value of minutes will be less than 24 X 60)

Input

Input is a single integer.

```
1180
```

Output

Output is a string denoting the digital clock time.

```
19:40
```

Solution: digitalClock.cpp

Biggest Number String

You are given a vector of numbers. You want to concatenate these numbers together to form the lexicographically largest number. Print that largest number number. You can't rearrange the digits of any number, however you can place the numbers next to each other in any order.

Input

```
10,11,20,30,3
```

Output

```
330201110
```

You can verify that this is the largest number that we can form.

Solution: biggestNumber.cpp

Run Length Encoding

Write a function to perform basic string compression using the counts of repeated characters, also known as Run Length Encoding. Let see one example, the input string "aaaabccccaaa" would become "a4b1c5a3". If the "compressed" string would not become smaller than the original string, your function should return the input string. You can assume the string has only uppercase and lowercase letters. You may use the `to_string(int)` method to convert an integer into string.

Sample Inputs

```
bbbaaaadexxxxxxabc
```

Sample Outputs

```
b3a4d1e1x6abc
```

Solution: runLenEncoding.cpp