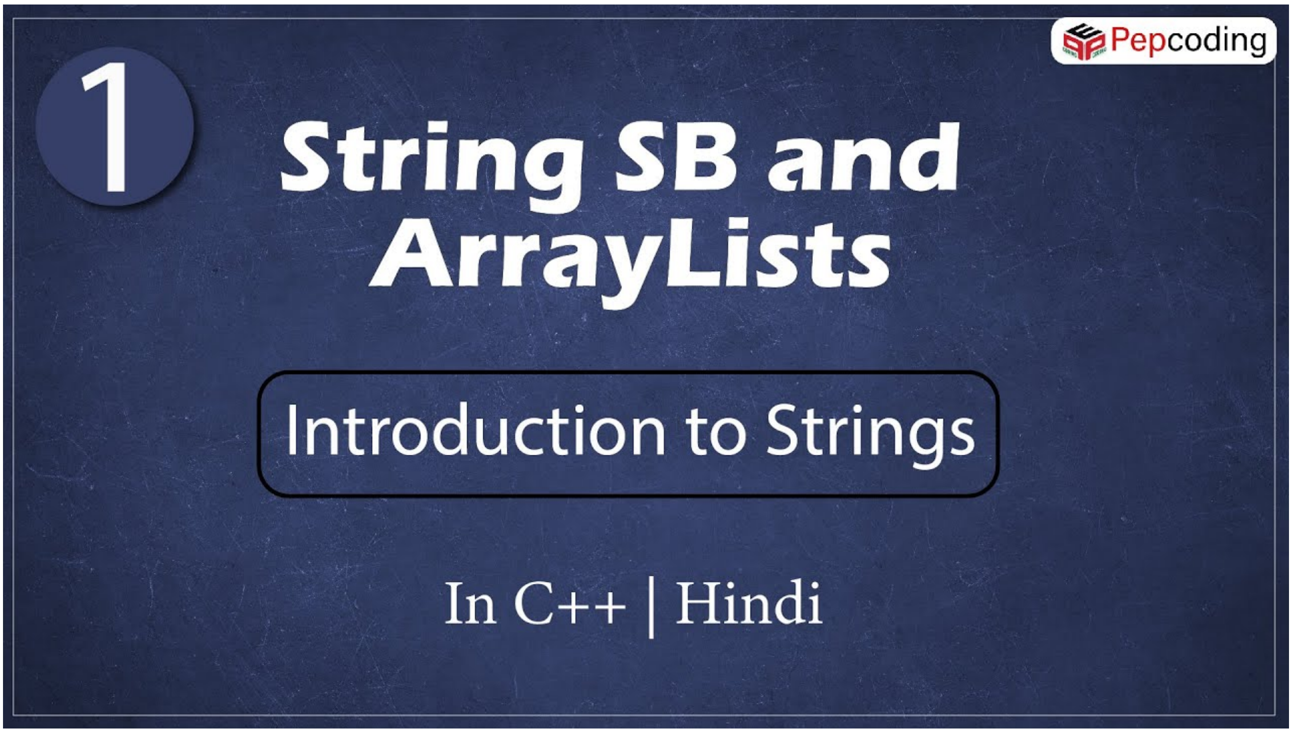
≠ string,dynamic string and dynamic array

Introduction to String



String SB and ArrayLists

Introduction to Strings

In C++ | Hindi

Print All Palindromic Substrings

Easy

1. You are given a string.
2. You have to print all palindromic substrings of the given string.

Constraints

```
1 <= length of string <= 500
```

Format

Input

A String

Output

All palindromic substrings(one in a line).
First, all palindromic substrings starting from first character of string will be printed, then from second character and so on.

Example

Sample Input

```
abcc
```

Sample Output

```
a
b
c
cc
c
```

```
#include <iostream>
#include <string>

using namespace std;

bool ispalindrome(string s){
    for (int i{};i<(s.length()/2);i++){
        if (s[i]!=s[s.length()-1-i]){
            return 0;
        }else{
            continue;
        }
    }
    return 1;

}

void printallpalindromess(string str){

    for (int i{};i<str.length();i++){
        for (int j{1};j<=(str.length()-i);j++){
            bool flag = ispalindrome(str.substr(i,j));
            if(flag == true){
                cout<<str.substr(i,j)<<endl;
            }
        }
    }

}


int main(){
    string str;
    getline(cin,str);
    printallpalindromess(str);

    return 0;
}
```

String Compression

Easy

1. You are given a string.
2. You have to compress the given string in the following two ways -

First compression -> The string should be compressed such that consecutive duplicates of characters are replaced with a single character.

For "aaabbccdee", the compressed string will be "abcde".

Second compression -> The string should be compressed such that consecutive duplicates of characters are replaced with the character and followed by the number of consecutive duplicates.

For "aaabbccdee", the compressed string will be "a3b2c2de2".

Constraints

```
1 <= length of string <= 1000
```

Format

Input

A String

Output

Two strings representing first compressed string and second compressed string respectively.

Example

Sample Input

```
wwwwaaadexxxxxx
```

Sample Output

```
wadex
w4a3dex6
```

```
#include<iostream>
#include<string.h>
using namespace std;


void String_compression01(string str)
{
   //write your code here

   //first element remain same
   int index {0};
   for(int i{1};i<str.length();i++){
       if (str[i]==str[i-1]){
           continue;
       }else{
           index++;
           str[index]=str[i];
       }
   }
   str.erase(index+1,str.length());
   cout<<str<<endl;
}
void String_compression02(string str){
```

```
//write your code here
//int i{};
int j{1};
int index{1};
 // while(str[i]!=0){


 // }

 for(int i{1};i<str.length();i++){
    if (str[i]==str[i-1]){
        j++;
        if(i==str.length()-1){
             //putting the number in the str
             //cout<<"end "<<endl;
             if(j>1){
                 if(j<10){
                  str[index]='0'+j;
                  index++;
                 }else{
                     string cn = to_string(j);
                     for(auto acn:cn){
                         str[index]=acn;
                         index++;
                     }

                  }
            }
        }

    }else{

        //putting the number in the str
        if(j>1){
           if(j<10){
            str[index]='0'+j;
            index++;
            }else{
                  string cn = to_string(j);
                  for(auto acn:cn){
                     str[index]=acn;
                     index++;
                  }

            }
        }
        str[index]=str[i];
        index++;
        j=1;
    }
    }
    if(index==1){
        cout<<str<<endl;
    }else{
        if(index!=str.length()){
            str.erase(index,str.length()-1);

        }
        cout<<str<<endl;

    }


}

int main()
{
    string str;
    getline(cin,str);
    String_compression01(str);
    String_compression02(str);
}
```

Toggle Optica Document

Easy

1. There is an agreement document for JSCOP optica but by mistake, the case of every character in the string of document is a toggle so you need to correct it.
2. You are given a string of the document that contains only lowercase and uppercase alphabets.
3. You have to toggle the case of every character of the given string.

Constraints

1 <= length of string <= 1000

Format

Input

A String

Output

A String

Example

Sample Input

`pepCODinG`

Sample Output

`PEPcodINg`

```
#include<iostream>
#include<string>
using namespace std;

void togglecase(string & str){

    for(auto &c:str){
        if(c>='a'&&c<='z'){
            c = 'A'+c-'a';
        }else if(c>='A'&&c<='Z'){
            c = 'a'+c-'A';
        }
    }
    return ;
}

int main() {
  string str;
  getline(cin, str);
  togglecase(str);
  cout <<str<<endl;
  return 0;
}
```

String With Difference Of Every Two Consecutive Characters

Easy

1. You are given a string that contains only lowercase and uppercase alphabets.
2. You have to form a string that contains the difference of ASCII values of every two consecutive characters between those characters.

For "abecd", the answer should be "a1b3e-2c1d", as

'b'-'a' = 1

'e'-'b' = 3

'c'-'e' = -2

'd'-'c' = 1

Constraints

```
1 <= length of string <= 1000
```

Format

Input

A String

Output

A String

Example

Sample Input

```
pepCODinG
```

Sample Output

```
p-11e11p-45C12O-11D37i5n-39G
```

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

using namespace std;

void printstr(string str)
{
   //write your code here

    int len = str.length();
    string new_string{str[0]};
    for(int i{1};i<len;i++){
        int diff{};
        diff = (str[i] - str[i-1]);
        string ds = to_string(diff);
        new_string += (ds+str[i]);
    }
    cout<<new_string<<endl;
}
```

```
int main()
{
    string str;
    getline(cin,str);
    printstr(str);
    return 0;
}
```

Remove Primes

Easy

1. You are given an ArrayList of positive integers. 2. You have to remove prime numbers from the given ArrayList and return the updated ArrayList. Note -> The order of elements should remain same.

Constraints

1 <= N <= 10000

Format

Input

A number N arr1 arr2.. N numbers

Output

An Arraylist

Example

Sample Input

```
4
3 12 13 15
```

Sample Output

```
[12, 15]
```

```
#include<iostream>
#include<vector>
#include<cmath>

using namespace std;

bool is_prime (int n){
    for (int i{2};i<=sqrt(n);i++){

        if((n%i)==0){

            return 0;
        }
    }
    return 1;
}
void removeprimes(vector<int> &v)
{
    //write your code here
    int s =v.size();
    for (int i{s-1};i>=0;i--){
        if(is_prime(v[i])){
            v.erase(v.begin()+i);
            //cout<<"GGHH"<<endl;
        }
        //cout<<"dgdfg"<<endl;
    }
    return;
}
int main()
{
    int n;
    cin>>n;
    vector<int> v(n,0);
    for(int i=0;i<n;i++)
    {
        cin>>v[i];
    }
    removeprimes(v);
    cout<<"[";
    for(int i {};i<v.size();i++ ){
        cout<<v[i];
        if(i==(v.size()-1)){
            break;
        }else{
            cout<<", ";
        }
       // cout<<"dddddd"<<endl;
    }
    cout<<"]";
}
```

Print All Permutations Of A String Iteratively

Easy

1. You are given a string. 2. You have to print all permutations of the given string iteratively.

Constraints

```
1 <= length of string <= 15
```

Format

Input

A String

Output

All permutations of the given string(one in a line).

Example

Sample Input

```
abc
```

Sample Output

```
abc
bac
cab
acb
bca
cba
```

```
#include<iostream>
#include<string>
using namespace std;

int facto(int n){
    if(n==0||n==1){
        return 1;
    }else{
        return n*facto(n-1);
    }
}

void printallperm(string str)
{
    //write your code here
    int s = str.size();
    int nop = facto(s);
    int index {};
    int quo{};
    for(int i{};i<nop;i++){
        string st =str;
        quo =i;
        for(int j{s};j>0;j--){
            index = quo%j;
            quo = quo/j;

            cout<<st[index];
            st.erase(index,1);
        }
        cout<<endl;
    }
}
int main()
{
 string str;
 getline(cin,str);

 printallperm(str);
}
```