**<https://www.youtube.com/watch?v=epBsoPWOySk&list=PLoa_roVVsxA1CJu4DsOljb9d7FJkMCynM&index=4>**

**[L1]Class - 4 ( Discussion on Practice problems of STL String ) [ Bangla ]**

**Class Video link :**

<https://youtu.be/epBsoPWOySk>

**Discussed topics :**

1) Some built in functions & Practice problems

Full code of string : <https://pastebin.com/x18NfqLp>

2) <https://codeforces.com/problemset/problem/118/A>

My Code : <https://codeforces.com/contest/118/submission/87687803>

3) <https://www.hackerearth.com/problem/algorithm/the-palindrome-2/description/>

My Code : <https://paste.ubuntu.com/p/VSsKZNcMSW/>

4) <https://practice.geeksforgeeks.org/problems/sum-of-digit-is-pallindrome-or-not/0>

My Code : <https://paste.ubuntu.com/p/W3kNx9Tps2/>

5) <https://codeforces.com/contest/112/problem/A>

My Code : <https://codeforces.com/contest/112/submission/87690360>

6) <https://codeforces.com/contest/78/problem/A>

My Code : <https://codeforces.com/contest/78/submission/87691840>

7) <https://www.hackerearth.com/problem/algorithm/lexogeek-e6450fb7/description/>

My Code : <https://paste.ubuntu.com/p/HTxQs7gmV8/>

**Class Tasks :**

1)<https://codeforces.com/problemset/problem/118/A>

2) <https://www.hackerearth.com/problem/algorithm/the-palindrome-2/description/>

3) <https://practice.geeksforgeeks.org/problems/sum-of-digit-is-pallindrome-or-not/0>

4) <https://codeforces.com/contest/112/problem/A>

5) <https://codeforces.com/contest/78/problem/A>

6) <https://www.hackerearth.com/problem/algorithm/lexogeek-e6450fb7/description/>

7) <http://www.lightoj.com/volume_showproblem.php?problem=1023>

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

*#define endl '\n'*

int main()

{

optimize();

vector<string>v;

v.push\_back("hamid");

v.push\_back("hamid");

v.push\_back("azad");

v.push\_back("azad");

v.push\_back("hosen");

v.push\_back("hosen");

sort(v.begin(),v.end());

int size=unique(v.begin(),v.end())-v.begin();

cout<<size<<endl;

**for**(int i=0;i<size;i++) cout<<v[i]<<endl;

**return** 0;

}

**Output:**

3

azad

hamid

hosen

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

*#define endl '\n'*

int main()

{

optimize();

char c='A';

bool f1=isupper(c);

cout<<f1<<endl;

**return** 0;

}

**Output:**

1

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

*#define endl '\n'*

int main()

{

optimize();

char c=' ';

bool f1=isspace(c);

cout<<f1<<endl;

**return** 0;

}

**Output:**

1

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

*#define endl '\n'*

int main()

{

optimize();

char c='a';

c=toupper(c);

//c=tolower(c);

cout<<c<<endl;

**return** 0;

}

**Output:**

A

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

*#define endl '\n'*

int main()

{

optimize();

string s="Hamid ";

string s1="Hosen";

copy(s1.begin(),s1.end(),back\_inserter(s));

cout<<s<<endl;

**return** 0;

}

**Output:**

Hamid Hosen

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

*#define endl '\n'*

int main()

{

optimize();

string s="absaabassaaa";

s.erase(remove(s.begin(),s.end(),'a'),s.end());

cout<<s<<endl;

**return** 0;

}

**Output:**

bsbss

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

*#define endl '\n'*

int main()

{

optimize();

string s="Hamid";

**if**(s.find("mid")!=-1) cout<<"Sub string found"<<endl;

**else** cout<<"Not found"<<endl;

**return** 0;

}

**Output:**

Sub string found

///In the name of ALLAH

*#include<bits/stdc++.h>*

using namespace std;

int main ()

{

/// Declare string

string s;

/// Assign string

s = "abcdf";

/// Printing size of string

cout << s.size() << endl; /// 5

/// Printing string

cout << s << endl; /// abcdf

/// Pushing char back to a string

s += 'b';

s += 'c';

cout << s << endl; /// abcdfbc

/// Taking input string

cin >> s;

cout << s << endl;

s = "asdfgg";

/// Checking **is** a string empty **or** **not**

string s1;

cout << s.empty() << endl; /// 0

cout << s1.empty() << endl; /// 1

/// Assigning an string **in** another string variable

s1 = s;

s.clear();

cout << s.empty() << endl; /// 1

cout << s1.empty() << endl; /// 0

/// assigning 'k' **in** 0-th index

s = "asdfg";

s[0] = 'k';

cout << s << endl; ///kasdfg;

s = "abc";

s1 = "def";

/// String concatenation

string tmp = s + s1;

cout << tmp << endl; /// abcdef

/// String iterator

string::iterator it;

**for** ( it = s.begin(); it != s.end(); it++ ) cout << \*it; /// abc

cout << endl;

/// For each loop

**for** ( auto c : s ) cout << c; /// abc

cout << endl;

s = "asd";

tmp = s;

/// Comparing two strings

**if** ( tmp == s ) cout << "Yes Match**\n**";

**else** "No Match**\n**";

/// String reverse **and** checking **is** a string **is** palindrome **or** **not**

s = "asddsa";

tmp = s;

reverse( tmp.begin(), tmp.end() );

**if** ( tmp == s ) cout << "Yes Palindrome" << endl;

**else** cout << "Not Palindrome" << endl;

/// String sorting **in** non-decreasing order

s = "gfds";

sort ( s.begin(), s.end() );

cout << s << endl; /// dfgs

/// String sorting **in** non-increasing order

sort ( s.rbegin(), s.rend() );

cout << s << endl; /// sgfd

/// Getting all unique elements of a string. Be care full, string should be sorted.

s = "aaadddsss";

int n = unique( s.begin(), s.end() ) - s.begin();

**for** ( int i = 0; i < n; i++ ) cout << s[i];/// ads

cout << endl;

/// Getting maximum element of string

cout << \*max\_element( s.begin(), s.end() ) << endl; /// s

/// Getting minimum element of string

cout << \*min\_element( s.begin(), s.end() ) << endl; /// a

/// When we want to take input **with** space

/// input : Muhammad Shahriar Alam

char c;

cin >> c;

getline( cin, s );

s = c + s;

cout << s << endl; /// Muhammad Shahriar Alam

/// If we need to sort some string on lexicographical order :

vector<string> v;

v.push\_back( "Muhammad" );

v.push\_back( "Nova" );

v.push\_back( "Maslenia Mubarrat" );

v.push\_back( "CPS Academy" );

v.push\_back( "Rashedul Alam Anik" );

v.push\_back( "Farhan sadik Sakib" );

v.push\_back( "Gazi Mohaimin Iqbal" );

sort ( v.begin(), v.end() );

**for** ( auto u : v ) cout << u << endl;

/\*\*

**Output:**

CPS Academy

Farhan sadik Sakib

Gazi Mohaimin Iqbal

Maslenia Mubarrat

Muhammad

Nova

Rashedul Alam Anik

\*/

s = "asdf";

s.pop\_back(); /// removes last char of string

cout << s.back() << endl; /// print last char of string

v.clear();

v = { "Shahriar", "Shahriar", "Momo", "Momo", "Sharif", "Sharif" };

int Sz = unique ( v.begin(), v.end() ) - v.begin();

cout << Sz << endl; /// Number of unique strings **in** vector v;

**for** ( int i = 0; i < Sz; i++ ) cout << v[i] << endl; /// Prints all unique strings **in** vector v

/// Converting int to string

int a = 123;

s = to\_string (a);

cout << s << endl; /// 123

s[0] = '3';

cout << s << endl; /// 323

/// Converting string to integer

s = "123";

a = stoi ( s );

cout << a << endl; /// 123

a++;

cout << a << endl; /// 124;

/// Deleting a substring **from** **string**

s = "ShaKAKAhriar";

s.erase ( s.begin()+3, s.begin()+7 ); /// erase substring "KAKA" **from** **string** s

cout << s << endl;

/// Copying a substring of a string to a string

tmp = "Gagha Alam Gadha";

s = "Shahriar ";

copy ( tmp.begin()+6, tmp.begin()+10, back\_inserter ( s ) ); /// copying "Alam substring to string s back.

cout << s << endl; /// Shahriar Alam

/// Erasing all occurrence of a specific char **from** **string.**

s = "aaassdddaaasdd";

s.erase ( remove ( s.begin(), s.end(), 'a' ), s.end() ); /// removes all 'a' **from** **s**

cout << s << endl;

/// Checking **is** a string **is** substring of another string **in** O(n\*m)

s = "ashshasdakks";

**if** ( s.find( "asd" ) != -1 ) cout << "Substring found";

**else** cout << "Not found";

**return** 0;

}

**problem link=**[**https://codeforces.com/problemset/problem/118/A**](https://codeforces.com/problemset/problem/118/A)

**code=**[**https://codeforces.com/contest/118/submission/87687803**](https://codeforces.com/contest/118/submission/87687803)

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

*#define endl '\n'*

bool isVowel(char c){

**return** (c=='a'||c=='e'||c=='i'||c=='o'||c=='u'||c=='y');

}

int main()

{

optimize();

string s,s1;

cin>>s;

**for** (auto u:s){

char c=tolower(u);

**if**(isVowel(c)==0){

s1+='.';

s1+=c;

}

}

cout<<s1<<endl;

**return** 0;

}

**Output:**

Codeforces

.c.d.f.r.c.s

**problem link=**[**https://www.hackerearth.com/problem/algorithm/the-palindrome-2/description/**](https://www.hackerearth.com/problem/algorithm/the-palindrome-2/description/)

**code =**[**https://paste.ubuntu.com/p/VSsKZNcMSW/**](https://paste.ubuntu.com/p/VSsKZNcMSW/)

*#include <bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

*#define endl '\n'*

int main()

{

optimize();

int t;

string s,s1;

cin>>t;

**while** (t--){

cin>>s;

s1=s;

reverse(s1.begin(),s1.end());

**if**(s==s1) cout<<"Yes"<<endl;

**else** cout<<"No"<<endl;

}

**return** 0;

}

**Output:**

4

madam

ab

abba

a

Yes

No

Yes

Yes

**problem link=**[**https://practice.geeksforgeeks.org/problems/sum-of-digit-is-pallindrome-or-not2751/1**](https://practice.geeksforgeeks.org/problems/sum-of-digit-is-pallindrome-or-not2751/1)

**code=**[**https://paste.ubuntu.com/p/W3kNx9Tps2/**](https://paste.ubuntu.com/p/W3kNx9Tps2/)

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

int main()

{

optimize();

int t;

cin >> t;

**while** ( t-- ) {

string s;

cin >> s;

int digitSum = 0;

**for** ( auto u : s ) {

digitSum += ( u - '0' );

}

string str = to\_string ( digitSum );

string tmp = str;

reverse ( tmp.begin(), tmp.end() );

**if** ( tmp == str ) cout << "YES**\n**";

**else** cout << "NO**\n**";

}

**return** 0;

}

**Output:**

2

56

34

YES

YES

**problem=1**[**https://codeforces.com/contest/112/problem/A**](https://codeforces.com/contest/112/problem/A)

[**https://codeforces.com/contest/112/submission/87690360**](https://codeforces.com/contest/112/submission/87690360)

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

*#define elndl '\n'*

int main()

{

optimize();

string s1, s2;

cin >> s1 >> s2;

**for** ( int i = 0; i < s1.size(); i++ ) s1[i] = tolower ( s1[i] );

**for** ( int i = 0; i < s2.size(); i++ ) s2[i] = tolower ( s2[i] );

**if** ( s1 < s2 ) cout << "-1"<<endl;

**else** **if** ( s1 > s2 ) cout << "1"<<endl;

**else** cout << "0"<<endl;

**return** 0;

}

**Output:**

hamid

hosen

-1

**problem link=**[**https://codeforces.com/contest/78/problem/A**](https://codeforces.com/contest/78/problem/A)

**code=**[**https://codeforces.com/contest/78/submission/87691840**](https://codeforces.com/contest/78/submission/87691840)

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

bool isVowel ( char c )

{

**return** ( c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' );

}

int main()

{

optimize();

vector<string> v;

int cnt[] = { 5, 7, 5 };

**for** ( int i = 0; i < 3; i++ ) {

string s;

char c;

cin >> c;

getline ( cin, s );

s = c + s;

v.push\_back ( s );

}

**for** ( int i = 0; i < 3; i++ ) {

int c = 0;

**for** ( auto u : v[i] ) {

**if** ( isVowel(u) ) c++;

}

**if** ( c != cnt[i] ) **return** cout << "NO**\n**", 0;

}

cout << "YES**\n**";

}

**Output:**

on codeforces

beta round **is** running

a rustling of keys

YES

**problem link=**[**https://www.hackerearth.com/problem/algorithm/lexogeek-e6450fb7/description/**](https://www.hackerearth.com/problem/algorithm/lexogeek-e6450fb7/description/)

**code=**[**https://paste.ubuntu.com/p/HTxQs7gmV8/**](https://paste.ubuntu.com/p/HTxQs7gmV8/)

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

*#define endl '\n'*

int main()

{

optimize();

int t;

string s;

cin>>t;

**while** (t--){

cin>>s;

**if**(next\_permutation(s.begin(),s.end())) cout<<s<<endl;

**else** cout<<"no answer"<<endl;

}

**return** 0;

}

**Output:**

3

abcd

acdb

dcba

abdc

adbc

no answer

**link=**[**http://www.lightoj.com/volume\_showproblem.php?problem=1023**](http://www.lightoj.com/volume_showproblem.php?problem=1023)

*#include<bits/stdc++.h>*

using namespace std;

*#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

//*#define endl '\n'*

int main()

{

optimize();

int t,t1=1;

cin>>t;

**while** (t--){

cout<<"Case "<<t1<<":"<<endl;

string s="";

char c='A';

int x,y,count=0;

cin>>x>>y;

**for**(int i=0;i<x;i++){

s=s+c;

c++;

}

do {

cout<<s;

cout<<endl;

count++;

**if**(count==y) **break**;

}**while** (next\_permutation(s.begin(),s.end()));

t1++;

}

**return** 0;

}

**Output:**

2

Case 1:

3 8

ABC

ACB

BAC

BCA

CAB

CBA

Case 2:

10 10

ABCDEFGHIJ

ABCDEFGHJI

ABCDEFGIHJ

ABCDEFGIJH

ABCDEFGJHI

ABCDEFGJIH

ABCDEFHGIJ

ABCDEFHGJI

ABCDEFHIGJ

ABCDEFHIJG