**Reversing the vowels**

**Method1**

string modify (string s)

{

//code here.

int a = 0, b = s.size()-1;

vector<char> vowel {'a','e','i','o','u'};

while(b>=a) {

auto ita = find(vowel.begin(), vowel.end(), s[a]);

auto itb = find(vowel.begin(), vowel.end(), s[b]);

if(ita != vowel.end() && itb != vowel.end()) {

swap(s[a], s[b]);

a++;

b--;

}

else if(ita != vowel.end()) b--;

else a++;

}

return s;

}

**Method 2 using map**

string modify (string a)

{

//code here.

map<char,int> mp;

mp['a']++;mp['e']++;mp['i']++;mp['o']++;mp['u']++;

int s=0, e=a.size()-1;

while(s<e)

{

if(mp.find(a[s])!=mp.end() && mp.find(a[e])!=mp.end()) //present

{

swap(a[s],a[e]);

s++;e--;

}

if(mp.find(a[s])==mp.end()) // not present

s++;

if(mp.find(a[e])==mp.end()) // not present

e--;

}

return a;

}

**Method 3**

string modify (string S)

{

//code here.

string vow;

int i;

for( i=0;i<S.size();i++)

{

if(S[i]=='a' || S[i]=='e' || S[i]=='i' || S[i]=='o' || S[i]=='u')

{

vow = vow + S[i];

}

}

int n = vow.size();

for(i=0;i<S.size();i++)

{

if(S[i]=='a' || S[i]=='e' || S[i]=='i' || S[i]=='o' || S[i]=='u')

{

S[i]=vow[n-1];

n--;

}

}

return S;

}

**Method 4 Using stack**

string modify (string s)

{

//code here.

stack<char> vowels ;

for(int i = 0 ; i < s.size() ; i++)

if(s[i] == 'a' || s[i] == 'e' || s[i] == 'i' || s[i] == 'o' || s[i] == 'u')

vowels.push(s[i]) ;

for(int i = 0 ; i < s.size() ; i++)

if(s[i] == 'a' || s[i] == 'e' || s[i] == 'i' || s[i] == 'o' || s[i] == 'u')

{

s[i] = vowels.top() ;

vowels.pop() ;

}

return s ;

}

**Method 5**

bool isvowel(char ch){

if(ch =='a' || ch =='e' || ch =='i' || ch =='o' || ch =='u'){

return true;

}

return false;

}

string modify (string s)

{

//code here.

int i=0,j=s.size()-1;

while(i<j){

if(isvowel(s[i]) && isvowel(s[j])){

swap(s[i],s[j]);

i++;

j--;

}

if(!isvowel(s[i])){

i++;

}

if(!isvowel(s[j])){

j--;

}

}

return s;

}