**Q1:**

#include <iostream>

using namespace std;

string smallestStr(string str, int n)

{

int i, j;

int chk[MAX];

for (i = 0; i < MAX; i++)

chk[i] = -1;

for (i = 0; i < n; i++) {

if (chk[str[i] - 'a'] == -1)

chk[str[i] - 'a'] = i;

}

for (i = 0; i < n; i++) {

bool flag = false;

for (j = 0; j < str[i] - 'a'; j++) {

if (chk[j] > chk[str[i] - 'a']) {

flag = true;

break;

}

}

if (flag)

break;

}

if (i < n) {

char ch1 = str[i];

char ch2 = char(j + 'a');

// For every character

for (i = 0; i < n; i++) {

if (str[i] == ch1)

str[i] = ch2;

else if (str[i] == ch2)

str[i] = ch1;

}

}

return str;

}

int main()

{

string str = "ccad";

int n = str.length();

cout << smallestStr(str, n);

return 0;

}

**Output**

**Input:**

**A = "ccad"**

**Output: "aacd"**

**Q2:**

#include <algorithm>

#include <iostream>

#include <string>

#include <vector>

using namespace std;

int myCompare(string X, string Y)

{

string XY = X.append(Y);

string YX = Y.append(X);

return XY.compare(YX) > 0 ? 1 : 0;

}

void printLargest(vector<string> arr)

{

sort(arr.begin(), arr.end(), myCompare);

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

cout << arr[i];

}

int main()

{

vector<string> arr;

arr.push\_back("54");

arr.push\_back("546");

arr.push\_back("548");

arr.push\_back("60");

printLargest(arr);

return 0;

}

**Output:**

**Input:**

**N = 4**

**Arr[] = {54, 546, 548, 60}**

**Output: 6054854654**

**Q3:**

#include <iostream>

#include<conio.h>

using namespace std;

void printLPSS(char \*s)

{

int maxLength = 1;

int start = 0;

int n = strlen(s);

int low, high;

for (int i = 1; i < n; ++i)

{

low = i - 1;

high = i + 1;

while (low >= 0 && high < n && s[low] == s[high])

{

if (high - low + 1 > maxLength)

{

start = low;

maxLength = high - low + 1;

}

--low;

++high;

}

high = i;

while (low >= 0 && high < n && s[low] == s[high])

{

if (high - low + 1 > maxLength)

{

start = low;

maxLength = high - low + 1;

}

--low;

++high;

}

}

int end = start + maxLength - 1;

cout<<"Longest palindrome substring is ";

for( int i = start; i <= end; ++i )

{

cout<<s[i];

}

cout<<endl;

cout<<maxLength<<endl;

}

int main()

{

char s[] = "aaaabbaa";

printLPSS(s);

return 0;

}

**OUTPUT:**

**Input:**

**S = "aaaabbaa"**

**Output: aabbaa**

**Q4:**

#include <iostream>

#include<conio.h>

using namespace std;

struct meeting {

int start;

int end;

int pos;

};

bool endSort(struct meeting m1, meeting m2)

{

return (m1.end < m2.end);

}

void maximumMeetings(int s[], int f[], int n)

{

struct meeting meet[n];

int c = 1;

for (int i = 0; i < n; i++)

{

meet[i].start = s[i];

meet[i].end = f[i];

meet[i].pos = i + 1;

}

sort(meet, meet + n, endSort);

vector<int> m;

m.push\_back(meet[0].pos);

int prev\_end = meet[0].end;

for (int i = 1; i < n; i++) {

if (meet[i].start >= prev\_end)

{

m.push\_back(meet[i].pos);

prev\_end = meet[i].end;

c++;

}

}

cout<<"Maximum meetings that can take place are: "<<c;

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

cout << m[i] << " ";

}

}

int main()

{

int s[] = {1,3,0,5,8,5}

int f[] = { 2,4,6,7,9,9};

int n = sizeof(s) / sizeof(s[0]);

maximumMeetings(s, f, n);

return 0;

}

}

return count;

}

**Output:**

**Maximum meetings that can take place are: 4**