

Assignment #01

Data Structure and Algorithms.

Topic :

String data structure.

Submitted By : Muhammad Usman

FA21-BCS-020 (4-A).

Submitted To : Ma'am Tehreem.

Date: 21 March, 2023.

QUESTION #01

Write down c/c++ codes for the following functions of strings data structure:

Part (a)

Remove punctuations from a given string.

Code:

```
#include <iostream>
using namespace std;
int main() {
    cout << "Enter a string.";
    string str;
    getline(cin, str);
    for (int i = 0; i < str.length(); i++) {
        int num = (int) str[i];
        if ((num >= 33 & num <= 64) || (num >= 91 & num <= 96) ||
            (num >= 123 & num <= 126)) {
            str.erase(i, 1);
        }
    }
    cout << str;
    return 0;
}
```

Part (b)

Rearrange characters in a string such that no two adjacent are same

Code:

```
#include <iostream>
using namespace std;
int main() {
```

```
    cout << "Enter a string.";
```

```
    string str;
```

```
    getline (cin, str);
```

```
    for (int i=0; i < str.length()-1; ++i) {
```

```
        if (str[i] == str[i+1]) {
```

```
            char temp = str[i+1];
```

```
            str[i+1] = str[str.length()-1];
```

```
            str[str.length()-1] = temp; } }
```

```
    cout << str;
```

```
    return 0; }
```

Part (c)

Program to check if input is an integer or a string.

Code:

```
#include <iostream>
```

```
using namespace std;
```

```
bool Is-Number (string str) {
```

```
    int start = 0;
```

```
    if (str[start] == '+' || str[start] == '-') {
```

```
        start++; }
```

```
    for ( ; start < str.length(); start++) {
```



```

if (str[start] < '0' & str[start] > '9') {
    return false; } }
return true; }

```

```

int main() {
    cout << "Enter a string.";
    string str;
    getline (cin, str);
    if (Is_Number (str)) {
        cout << "Integer"; }
    else {
        cout << "String"; }
    return 0; }

```

Part (d)

C++ program to find second most frequent character.

Code:

```

#include <iostream>
using namespace std;
int main() {
    cout << "Enter a string.";
    string str;
    getline (cin, str);
    char alphabets[26];
    int repeated[26];

```

```

for (int i=0; i<26; i++) {
    repeated[i]=0;
    alphabets = 'a'+i;
    cout<<alphabets[i]<<" "; }

int count=0;
for (int i=0; i<26; i++) {
    count=0;
    for (int j=0; j<str.length(); j++) {
        if (alphabets[i] == str[j]) {
            count++; } }
    repeated[i]=count; }

for (int i=0; i<26; i++) {
    for (int j=0; j<26; j++) {
        if (repeated[i] > repeated[j]) {
            int temp = repeated[j];
            repeated[j] = repeated[i];
            repeated[i] = temp;
            char ch = alphabets[j];
            alphabets[j] = alphabets[i];
            alphabets[i] = ch; } } }

cout << "Second most frequent is:" << alphabets[1] << "number" << repeated[1]
return 0;
}

```


Part (c)

C++ program to sort an array of name or string

Code::

```
#include <iostream>
```

```
using namespace std;
```

```
int main () {
```

```
    cout << "How many names.";
```

```
    int size;
```

```
    cin >> size;
```

```
    string array[size];
```

```
    cout << "Enter " << size << " names";
```

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

```
        cin >> array[i];
```

```
    for (int i=0; i < size-1; i++) {
```

```
        for (int j=0; j < size; j++) {
```

```
            if (array[j] > array[j+1]) {
```

```
                string name = array[j+1];
```

```
                array[j+1] = array[j];
```

```
                array[j] = name; } } }
```

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

```
        cout << array[i] << " ";
```

```
    }
```

```
    return 0;
```

```
}
```

Part (f)

C++ program to concatenate a string given times.

Code:-

```
#include <iostream>
using namespace std;
int main() {
    cout << "Enter a string.";
    string str;
    getline(cin, str);
    cout << "How many times do you want to concatenate";
    int num;
    cin >> num;
    string str-1;
    for(int i=0; i<num; i++) {
        cout << "Enter a string " << i << "to concatenate";
        getline(cin, str-1);
        str = str.append(" " + str-1);
    }
    cout << "The original string is : " << str;
    return 0; }
```

Part (g)

In C++, there are several ways to add characters to a string object

std::string::append()

append is a member function of the std::string class that is used to add a sequence of characters to end of existing string.

Example:-

```
std::string str1 = "Hello";
```

```
std::string str2 = "World";
```

```
str1.append(str2);
```

After this str1 contain "Hello world".

std::string::push_back()

push_back() is a member function of the std::string class that is used to add a single character to the end of the existing string.

Example:-

```
std::string str1 = "Hello";
```

```
str1.push_back('!');
```

After this : str will contain "Hello!"

Operator +=

The += operator is an overloaded of the std::string class that is used to concatenate two strings.

Example:-

```
std::string str1 = "Hello";
```

```
std::string str2 = "World!";
```

```
str1 += str2;
```

After this str1 will contain "Hello world!"

Part (h)

Comparing two strings in C++

Code:

```
#include <iostream>
```

```
using namespace std;
```

```
int main () {
```

```
    cout << "Enter a strings. ";
```

```
    string str-1, str-2;
```

```
    getline (cin, str-1);
```

```
    getline (cin, str-2);
```

```
    int num-1 = str-1.length();
```

```
    int num-2 = str-2.length();
```

```
    if (num-1 != num-2) {
```

```
        cout << "The both string are not equal. "; }
```

```
    else {
```

```
        for (int i=0; i<str-1.length(); i++) {
```

```
            if (str-1[i] != str-2[i]) {
```

```
                cout << "The both string string are not equal";
```

```
                break; } }
```

```
        cout << "The both strings are equal. "; }
```

```
    return 0;
```

```
}
```

Part (i)

Extract all the integers from string in C++

Code:

```
#include <iostream>
using namespace std;
int main() {
    cout << "Enter a string: ";
    string str;
    cin >> str;
    string num-str = "";
    for (int start=0; start < str.length(); start++) {
        if (str[start] >= '0' & str[start] <= '9') {
            num-str += str[start];
        }
    }
    cout << num-str;
    return 0; }
```

Part (j)

C++ program to Replace a word in a text by another given word.

Code:

```
#include <iostream>
using namespace std;
int main() {
    cout << "Enter a string: ";
```



```

string str, str-split = "", str-array[100];
getline(cin, str);
int index = 0;
for (int i = 0; i < str.length(); i++) {
    if (str[i] != ' ') {
        str-split += str[i];
    }
    else {
        str-array[index] = str-split;
        str-split = "";
        index++;
    }
}
str-array[index] = str-split;
for (int i = 0; i <= index; i++) {
    cout << str-array[i] << " ";
}
cout << " Pock word";

string str-org, str-replace;
cin >> str-org;
cout << " Enter replace word.";
cin >> str-replace;
for (int i = 0; i < index; i++) {
    if (str-array[i] == str-org) {
        str-array[i] = str-replace;
    }
}
return 0;
}

```