Lab 12 – Strings

Task 01:

Imagine a greeting card company that allows customers to personalize their greeting cards with custom messages. One of the features provided by the company is the ability to replace specific words in their messages with other words. This is particularly useful for creating a consistent theme across multiple cards, such as changing "Birthday" to "Anniversary" or "Congratulations" to "Well Done". The company has tasked you with developing a software solution that will allow customers to easily make these changes to their messages.

Code:

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

int main() {
    string message, oldWord, newWord;
    cout << "Enter your message: ";
    getline(cin, message);
    cout << endl;

    cout << "Enter the word you want to replace: ";
    cin >> oldWord;
    cout << "Enter the new word: ";
    cin >> newWord;

    size_t pos = message.find(oldWord); // size_t -> special sort of data type
which is used to store index or size of string
    if (pos == string::npos) { // means not found
        cout << "Word not found!" << endl;
    }
    else {
        message.replace(pos, oldWord.length(), newWord);
        cout << "\nUpdated Message: " << message << endl;
    }
}
```

Output:

```
PS D:\Hasan\cpp\00. university\lab12-Strings> g++ task01.cpp
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe
Enter your message: How are me?

Enter the word you want to replace: me
Enter the new word: you

Updated Message: How are you?
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe
Enter your message: hello!

Enter the word you want to replace: lo
Enter the new word: hi

Updated Message: helhi!
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe
Enter your message: bye

Enter the word you want to replace: ok
Enter the new word: hi
Word not found!
PS D:\Hasan\cpp\00. university\lab12-Strings>
```

Task 02:

You are working on a customer service platform where agents often need to combine various pieces of information to form a complete response to customer inquiries. To streamline this process, you will create a feature that allows agents to input different parts of their response separately and then concatenate them into a single, coherent message.

Develop a C++ program that allows customer service agents to input multiple parts of a response message, concatenate these parts into a single message, and display the final message.

Code:

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

int main() {
    string part1, part2, part3, part4, finalPart;

    cout << "Enter the first part of the response: ";
    getline(cin,part1);
    cout << "Enter the second part of the response: ";
    getline(cin, part2);
    cout << "Enter the third part of the response: ";
    getline(cin, part3);
    cout << "Enter the fourth part of the response: ";
    getline(cin, part4);

    finalPart = part1 + " " + part2 + " " + part3 + " " + part4;
    cout << "\nFinal Response: " << finalPart << endl;
}
```

Output:

```
PS D:\Hasan\cpp\00. university\lab12-Strings> g++ task02.cpp
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe
Enter the first part of the response: Thankyou
Enter the second part of the response: for
Enter the third part of the response: your
Enter the fourth part of the response: visit.

Final Response: Thankyou for your visit.
PS D:\Hasan\cpp\00. university\lab12-Strings>
```

Task 03:

Imagine we're developing a program tailored for creating user accounts, where the system necessitates specific string manipulations to ensure the usernames adhere to the required format.

- Convert the string to uppercase.
- Replace all spaces with underscores ('_').
- Reverse the string.
- Return the modified string.

Code:

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

int main() {
    string username;
    cout << "Enter a username: ";
    getline(cin, username);

    // first converting all characters in uppercase
    for (int i = 0; i < username.length(); i++) {
        if (username[i] > 96 && username[i] < 123) {
            username[i] -= 32;
        }
    }

    // replacing space with _
    for (int i = 0; i < username.length(); i++) {
        if (username[i] == ' ') {
            username[i] = '_';
        }
    }

    // flipping them
    int start = 0, end = username.length() - 1;
    while (start < end) {
        char temp = username[start];
        username[start] = username[end];
        username[end] = temp;
        start++;
        end--;
    }
    cout << "\nModified Username: " << username << endl;
}
```

Output:

```
PS D:\Hasan\cpp\00. university\lab12-Strings> g++ task03.cpp
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe
Enter a username: Muhmmad Hasan

Modified Username: NASAH_DAMMHUM
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe
```

Task 04

Write a program that takes a paragraph as input and formats it into a well-structured document. The program should capitalize the first letter of each sentence, ensure proper spacing between words, and handle special cases such as abbreviations and proper nouns.

Requirements:

- Prompt the user to enter a sentence.
- Read the input sentence from the user.
- Convert the first letter of each word to uppercase.
- Print the modified sentence with the capitalized words.

Code:

```
#include<iostream>
#include<string>
using namespace std;
int main() {
    string sent;
    cout << "Enter a sentence with four words: ";
    getline(cin, sent);

    size_t pos1 = sent.find(' ');
    size_t pos2 = sent.find(' ', pos1 + 1);
    size_t pos3 = sent.find(' ', pos2 + 1);
    string word[4] = {sent.substr(0, pos1), sent.substr(pos1+1, (pos2-pos1)-1),
sent.substr(pos2+1, (pos3-pos2)-1), sent.substr(pos3+1)};

    for(int i = 0; i < 4; i++){
        if(word[i][0] >= 97 && word[i][0] <= 122){
            word[i][0] -= 32;
        }
    }
    cout << "\nFormatted Sent: ";
    for(int i = 0; i < 4; i++){
        cout << word[i] << " ";
    }
    cout << endl;
}
```

Output:

```
PS D:\Hasan\cpp\00. university\lab12-Strings> g++ task04.cpp
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe
Enter a sentence with four words: My name is Hasan

Formatted Sent: My Name Is Hasan
```