

**OBJECTIVES**

To help the students in learning the concepts of strings in C++.

* To help the students in implementing and handling strings operations on string
* with the help on various examples.

**OUTCOMES**

After completing this, the students would be able to:

* Handle strings manipulation in the programs.
* Develop & implement simple real-life examples of strings/string functions.

**PROBLEMS**

1# Write a C++ program that uses functions to perform the following operations:

i. To insert a sub string into a given main string from a given position.

ii. To delete n characters from a given position in a given string.

#include <iostream>

#include <string>

using namespace std;

string insertSubstring(string mainStr, string subStr, int position) {

return mainStr.insert(position, subStr);

}

string deleteCharacters(string str, int position, int n) {

return str.erase(position, n);

}

int main() {

string mainString, subString;

int insertPosition, deletePosition, numCharsToDelete;

cout << "Enter the main string: ";

getline(cin, mainString);

cout << "Enter the substring to insert: ";

getline(cin, subString);

cout << "Enter the position to insert the substring: ";

cin >> insertPosition;

string resultStringAfterInsertion = insertSubstring(mainString, subString, insertPosition);

cout << "Result after inserting substring: " << resultStringAfterInsertion << endl;

cout << "Enter the position to start deletion: ";

cin >> deletePosition;

cout << "Enter the number of characters to delete: ";

cin >> numCharsToDelete;

string resultStringAfterDeletion = deleteCharacters(resultStringAfterInsertion, deletePosition, numCharsToDelete);

cout << "Result after deleting characters: " << resultStringAfterDeletion << endl;

return 0;

}

2# Write a C++ program to determine if the given string is a palindrome or not.

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

bool isPalindrome(const string& str) {

string reversedStr = str;

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

return str == reversedStr;

}

int main() {

string inputString;

cout << "Enter a string: ";

cin >> inputString;

if (isPalindrome(inputString)) {

cout << inputString << " is a palindrome." << endl;

} else {

cout << inputString << " is not a palindrome." << endl;

}

return 0;

}

3# Write a C++ program to find a string within a sentence and replace it with

another string.

#include <iostream>

#include <string>

using namespace std;

int main() {

string sentence, findStr, replaceStr;

cout << "Enter a sentence: ";

getline(cin, sentence);

cout << "Enter the string to find: ";

cin >> findStr;

cout << "Enter the string to replace it with: ";

cin >> replaceStr;

size\_t pos = sentence.find(findStr);

while (pos != string::npos) {

sentence.replace(pos, findStr.length(), replaceStr);

pos = sentence.find(findStr, pos + replaceStr.length());

}

cout << "Modified sentence: " << sentence << endl;

return 0;

}

4# Write a C++ program that reads a line of text and counts all occurrence of a

particular word.

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

int main() {

string line, word;

cout << "Enter a line of text: ";

getline(cin, line);

cout << "Enter the word to count: ";

cin >> word;

int count = 0;

string temp;

stringstream ss(line);

while (ss >> temp) {

if (temp == word) {

count++;

}

}

cout << "The word '" << word << "' occurs " << count << " times in the given line of text." << endl;

return 0;

}

5# Write a C++ program that displays the position or index in the string S where

the string T begins, or 1 if S doesn’t contain T.

#include <iostream>

#include <string>

using namespace std;

int main() {

string S, T;

cout << "Enter the main string (S): ";

getline(cin, S);

cout << "Enter the substring to find (T): ";

getline(cin, T);

size\_t position = S.find(T);

if (position != string::npos) {

cout << "Substring T begins at position " << position + 1 << " in the main string S." << endl;

} else {

cout << "Substring T is not found in the main string S. Returning position 1." << endl;

}

return 0;

}