

C++ Assignments | Strings - 1 | Week 7

1. Input a string of size n and update all the odd positions in the string to character '#'. Consider 0-based indexing.

Input : str = "Pbweshkuighlds"

Output : "P#w#s#k#i#l#l#s"

input : str = "a"

Output : "a"

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

```
string updateOddPositions(string str) {
    for (int i = 1; i < str.length(); i += 2) {
        str[i] = '#';
    }
    return str;
}
```

```
int main() {
    string str1 = "Pbweshkuighlds";
    string str2 = "a";

    cout << "Updated string 1: " << updateOddPositions(str1) << endl;
    cout << "Updated string 2: " << updateOddPositions(str2) << endl;

    return 0;
}
```

2. Input a string of length n and count all the consonants in the given string.

Input : "pwians"

Output : 4

Input : "abdc"

Output : 3

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

```
bool isVowel(char ch) {
    ch = tolower(ch);
    return (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u');
}
```

```
int countConsonants(const string& str) {
    int count = 0;
    for (char ch : str) {
```

```

        if (isalpha(ch) && !isVowel(ch)) {
            count++;
        }
    }
    return count;
}

int main() {
    string str1 = "pwians";
    string str2 = "abdc";

    cout << "Consonants in string 1: " << countConsonants(str1) << endl;
    cout << "Consonants in string 2: " << countConsonants(str2) << endl;

    return 0;
}

```

3.

Check whether the given string is palindrome or not.

Input : "abcde"

Output : No

Input : "abdcdba"

Output : Yes

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```

bool isPalindrome(const string& str) {
    int left = 0;
    int right = str.length() - 1;

    while (left < right) {
        if (str[left] != str[right]) {
            return false;
        }
        left++;
        right--;
    }
    return true;
}

```

```

int main() {
    string str1 = "abcde";
    string str2 = "abdcdba";

    cout << "Is string 1 palindrome? " << (isPalindrome(str1) ? "Yes" : "No") <<
endl;
}

```

```

    cout << "Is string 2 palindrome? " << (isPalindrome(str2) ? "Yes" : "No") <<
endl;

    return 0;
}

```

4.

Input a string of even length and reverse the second half of the string.

Input : str = "abcdefgh"

Output : abcdhgfe

Input :str = "pwians"

Output : pwisna

```
#include <iostream>
```

```
#include <string>
```

```
#include <algorithm>
```

```
using namespace std;
```

```

string reverseSecondHalf(string str) {
    int n = str.length();
    if (n % 2 == 0) {
        reverse(str.begin() + n / 2, str.end());
    }
    return str;
}

```

```

int main() {
    string str1 = "abcdefgh";
    string str2 = "pwians";

    cout << "Modified string 1: " << reverseSecondHalf(str1) << endl;
    cout << "Modified string 2: " << reverseSecondHalf(str2) << endl;

    return 0;
}

```

5.Input a string of length less than 10 and convert it into integer without using builtin function.

Input : "3244"

Output : 3244

Input : "12"

Output : 12

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```

int stringToInteger(const string& str) {
    int result = 0;
    for (char ch : str) {

```

```
        result = result * 10 + (ch - '0');  
    }  
    return result;  
}
```

```
int main() {  
    string str1 = "3244";  
    string str2 = "12";  
  
    cout << "Integer value of string 1: " << stringToInteger(str1) << endl;  
    cout << "Integer value of string 2: " << stringToInteger(str2) << endl;  
  
    return 0;  
}
```