



Assignment: 01

Course title: Data Structure Laboratory

Course Code: CSE 212

Submitted by,

Name: Mohammad
Fahim

ID: 242002112

Section: 6

Department: CSE

Submitted to,

Name: Mohammad
Akbar Bin Shah

Designation: Lecturer,
SoSET

Date Of Submission: 22-02-2025

Problem: 01

Maximum Number of Words Found in Sentences:

Source Code(c++):

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n;
    cout << "Enter the number of sentence: ";
    cin >> n;
    cin.ignore();

    int maxword = 0;

    for(int i=1; i<=n;i++)
    {
        string sentence;
        cout << "Enter sentence "<<i<<": ";
        getline(cin, sentence);

        int wordcount = 1;
        for(char ch : sentence)
        {
            if(ch == ' ')
            {
                wordcount++;
            }
        }
    }
```

```
    maxword = max(maxword, wordcount);  
}  
cout<<"Maxximum number of words found in sentence:  
"<<maxword<<endl;  
    return 0;  
}
```

Input/Output:

Output

```
Enter the number of sentence: 3  
Enter sentence 1: alice and bob love leetcode  
Enter sentence 2: i think so too  
Enter sentence 3: this is great thanks very much  
Maxximum number of words found in sentence: 6|
```

```
=== Code Execution Successful ===
```

Problem: 02

Camel Case:

Source Code(c++):

```
#include <bits/stdc++.h>

using namespace std;

string CamelCase(string str)
{
    string result;
    bool capitalizeNext = false;

    for (char c : str)
    {
        if (isalpha(c))
        {
            if (result.empty())
            {
                result += tolower(c);
            }
            else
            {
                result += capitalizeNext ? toupper(c) : tolower(c);
            }
            capitalizeNext = false;
        }
        else
        {

```

```
        capitalizeNext = true;
    }
}
return result;
}

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

    cout << "CamelCase output: " << CamelCase(input) << endl;
    return 0;
}
```

Input/Output:

Output

```
Enter a string: cats AND*Dogs-are Awesome
CamelCase output: catsAndDogsAreAwesome
```

```
=== Code Execution Successful ===
```

Problem: 03

Snake Case:

Source Code(c++):

```
#include <bits/stdc++.h>

using namespace std;

string SnakeCase(string str)
{
    string result;

    for (char c : str)
    {
        if (isalpha(c))
        {
            result += tolower(c);
        }
        else if (!result.empty() && result.back() != '_')
        {
            result += '_';
        }
    }

    if (!result.empty() && result.back() == '_')
    {
        result.pop_back();
    }
}
```

```
    return result;
}

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

    cout << "Snake Case: " << SnakeCase(input) << endl;

    return 0;
}
```

Input/Output:

Output

```
Enter a string: cats AND*Dogs-are Awesome
CamelCase output: catsAndDogsAreAwesome
```

```
=== Code Execution Successful ===
```

Problem: 04

Palindrome:

Source Code(c++):

```
#include <bits/stdc++.h>

using namespace std;

bool Palindrome(string str)
{
    string cleaned_str = "";

    for (char c : str)
    {
        if (isalpha(c))
        {
            cleaned_str += tolower(c);
        }
    }

    int left = 0, right = cleaned_str.length() - 1;

    while (left < right)
    {
        if (cleaned_str[left] != cleaned_str[right])
        {
            return false;
        }
        left++;
    }
```



```
        right--;  
    }  
    return true;  
}  
  
int main()  
{  
    string input;  
    cout << "Enter a string: ";  
    getline(cin, input);  
  
    if (Palindrome(input))  
    {  
        cout << "true" << endl;  
    }  
    else  
    {  
        cout << "false" << endl;  
    }  
    return 0;  
}
```

Input/Output:

```
Enter a string: never odd or even  
true
```

```
=== Code Execution Successful ===
```

```
Enter a string: eye  
true
```

```
=== Code Execution Successful ===
```

```
Enter a string: hello  
false
```

```
=== Code Execution Successful ===
```