

Lab 1: Task 1: Take a string input and output it.

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

int main() {
    string s;
    cin >> s;
    cout << "Output:"<<s;
}
```

Lab 1 : Task 2: Find the length of a string without using library function.

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

int main() {
    string s = "Raihan";
    int i = 0;
    while (s[i])
        i++;
    cout << "Length of the string: ";
    cout << i << endl;
}
```

Lab 1: Task 3: Concatenate Two Strings into one string without using library function.

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

    string s1 = "Tanvir ";
    string s2 = "Ahmed";
    string Name = s1+ s2;
    cout << Name;
}
```

Lab 1: Task 4: Reverse a string without using Library functions.

```
#include<stdio.h>
#include<string.h>
int main() {
    char s[100], temp;
    int i, j = 0;
    printf("\nEnter the string :");
    gets(s);
    for( i=0,j = strlen(s) - 1 ; i < j ; i++,j--){
        temp = s[i];
        s[i] = s[j];
        s[j] = temp;
    }
    printf("\n Reverse string is :%s", s);
}
```



```

        return (0);
    }

```

Lab 2: Task 1: Write a function to identify whether the given input is a digit or not

```

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

void number(string m)
{
    int j;

    for (j = 0; j < m.length(); j++)
    {
        if (48 <= m[j] && m[j] <= 57)
        {
            cout << "This is a digit";
        }
        else
        {
            cout << "Not a digit";
        }
    }
}

int main()
{
    string s;
    cout << "Enter something" << endl;
    cin >> s;
    number(s);
}

```

Lab 2: Task 2: Write a program to check whether the given input is a keyword or not

```

#include <iostream>
#include <string>
using namespace std;
void iskeyword(string s){

    string keys[32]={
        "auto", "double", "int", "struct", "break", "else", "long",
        "switch", "case", "enum", "register", "typedef", "char",
        "extern", "return", "union", "const", "float", "short",
        "unsigned", "continue", "for", "signed", "void", "default",
        "goto", "sizeof", "volatile", "do", "if", "static", "while"
    };
}

```



```

for (int i = 0; i < 32; i++) {
    if (s.find(keys[i]) != string::npos) {
        cout << "It is a keyword";
        return;
    }
}

cout << "Not a keyword";

}

```

```

int main() {

    string inputstr;
    cout << "Enter the string" << endl;
    getline(cin,inputstr);

    iskeyword(inputstr);

}

```

Lab 3: Task 1: From a statement output the Lexical analyzed form.

```

#include <iostream>
#include <string>
#include <bits/stdc++.h>

using namespace std;

int c;

void number(string m)
{
    int j;

    for(j=0;j<m.length();j++)
    {
        if(48<=m[j] && m[j]<=57) //asci code of 0-9
        {
            cout<<" "<<m<<">";

            return;
        }
    }
}

void alphabet(string x)
{

```



```
string alp[26]={ "a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z"};
string alp2[26]={ "A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y","Z"};
```

```
string s="id";
for(int n=0;n<26;n++)
{
    if(alp[n]==x || alp2[n]==x)
    {
        c++;
        cout<<" <"<<s<<" "<<c<<">";
        return;
    }
}
```

```
}
```

```
void keyword(string n)
{
    string
keyword[32]={ "if","else","auto","break","case","char","const","continue","do","double","enum","extern","float","for","got
o","int","long","register","return","short","signed","sizeof","static","struct","switch","typedef","union","unsigned","void","
volatile","while"};
    int i;
    for(i=0;i<32;i++)
    {
        if(keyword[i]==n)
        {

            cout<<" <"<<n<<">";

            return;
        }
    }
}
```

```
void operator1(string s)
{
    string op[7]={ "+","-","*","/","=","(",")"};
    for(int i=0;i<7;i++)
    {
        if(op[i]==s)
        {
            cout<<" <"<<s<<">";

            return;
        }
    }
}
```

```
void printing(string str)
{
    istringstream ss(str); //stream the string into different variable

    string word;
```



```

while (ss >> word)
{

    keyword(word);
    number(word);
    alphabet(word);
    operator1(word);
}
}

int main()
{

    string input;

    cout<<"Enter input:\n";
    getline(cin,input);

    printing(input);

    return 0;
}

```

Lab 4: Task 1: Check whether a statement is properly parenthesized or not.

```

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

    string input;
    int p=0;
    int i;
    int f1=0;

    cout<<"Enter input:\n";
    getline(cin,input); //full line as input

    int len = input.length();
    for(i=0;i<len;i++)
    {
        if(input[i]=='(')
        {
            p++;
        }
        else if(input[i]==')')
        {
            p--;
        }
    }
    if(p == 0)
    {

```



```

    cout<<"Input is properly parenthesized";
}
else
{
    cout<<"Input is not properly parenthesized";
}
return 0;
}

```

Lab 4: Task 2: Evaluate a mathematical Statement if the statement is correct.

```

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

    string input;
    int p=0;
    int i;
    int f1=0;

    cout<<"Enter input:\n";
    getline(cin,input); //full line as input

    int len = input.length();
    for(i=0;i<len;i++)
    {
        if(input[i]=='(')
        {
            p++;
        }
        else if(input[i]==')')
        {
            p--;
        }
    }
    if(p == 0)
    {
        cout<<"Input is properly parenthesized";
        f1=1;
    }
    else
    {
        cout<<"Input is not properly parenthesized";
        return 0;
    }
}

```



```

int f=0;
if(f1==1)
{

for(int j=0;j<len;j++)
{
    if(input[j]=='+'||input[j]=='-'||input[j]=='*'||input[j]=='/')
    {
        if(input[j+1]=='+'||input[j+1]=='-'||input[j+1]=='*'||input[j-
1]=='/'||input[j+1]=='('||input[j+1]==')')
        {
            cout<<"\n Invalid expression";
            f=1;
        }
    }
}
}
if(f==0)
{
    cout<<"\n Input is a valid expression.";
}

return 0;
}

```

Lab 5: Task 1: Check English grammar of a statement.

```

#include<stdio.h>
#include<string.h>
#include<stdbool.h>

```

```

bool checkSentence(char str[])
{
    // Calculate the length of the string.
    int len = strlen(str);

    if (str[0] < 'A' || str[0] > 'Z') // Check if the first character is uppercase or not in the sentence
        return false;

    if (str[len - 1] != '.') //Check if the last character is a full stop or not
        return false;

    int prev_state = 0, curr_state = 0;

    int index = 1;

    while (str[index])
    {
        if (str[index] >= 'A' && str[index] <= 'Z')

```



```

        curr_state = 0;

    else if (str[index] == ' ')
        curr_state = 1;

    else if (str[index] >= 'a' && str[index] <= 'z')
        curr_state = 2;

    else if (str[index] == '.')
        curr_state = 3;

    if (prev_state == curr_state && curr_state != 2) //Two continuous upper case characters are not
allowed
        return false;

    if (prev_state == 2 && curr_state == 0) //current state upper case previous state lower
case
        return false;

    if (curr_state == 3 && prev_state != 1) //current state . and previous " " na hoi
        return (str[index + 1] == '\0');

    index++;

    prev_state = curr_state;
}
return false;
}

int main()
{
    char *str[] = { "I love cinema.", "The vertex is S.",
                    "I am single.", "My name is KG.",
                    "I lovE cinema.", "GeeksQuiz. is a quiz site.",
                    "I love Geeksquiz and Geeksforgeeks.",
                    " You are my friend.", "I love cinema" };

    int str_size = sizeof(str) / sizeof(str[0]);
    int i = 0;
    for (i = 0; i < str_size; i++)
        checkSentence(str[i])? printf("\n%s\" is correct \n", str[i]):
                                printf("\n%s\" is incorrect \n", str[i]);

    return 0;
}

```


Lab 5: Task 2: Ab $A=a|E$ for this grammar check an input can be derived or not.