A

Term-Work

On

Compiler Design Lab (PCS-601)

Submitted in partial fulfillment of the requirement for the VI semester

B.Tech

By

Deepti Joshi

2061801

…

Faculty-in-Charge

Mr. Devesh Pandey

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS

2022-2023

STUDENT’S DECLARATION

I, Deepti Joshi hereby declare the work, which is being presented in the term-work, entitled “Compiler Design lab “in partial fulfillment of the requirement for the award of the degree **B,Tech** in the session **2022-2023**, is an authentic record of my own work carried out under the supervision of Mr. Devesh Pandey.

The matter embodied in this term-work has not been submitted by me for the award of any other degree.

Date: …………   ……………….

Full Signature of Student

CERTIFICATE

The term-work entitled "Compiler Design lab" being submitted by Deepti Joshi D/o Mr. Shekhranand Joshi enrollment no PV-B2061801 Roll no 2061801 to Graphic Era Hill University Bhimtal Campus for the award of bonafide work carried out by her. She has worked under my guidance and supervision and fulfilled the requirement for the submission of report.

(Mr. Devesh Pandey)	**(Dr. Ankur Bisht)**
Faculty-in-Charge	**(HOD, CSE Dept.)**

Practical – 1

Objective- Write a program to count tokens in an expression.

```
#include<bits/stdc++.h>
using namespace std;
bool isDelimiter(char ch)
{
   if(ch == ' ' || ch == '+' || ch == '-' || ch == '*' || ch == '/' || ch == '(' || ch == ')' || ch == '{' || ch
== '}' || ch == '[' || ch == ']'|| ch == '='|| ch == '=')
   return true;
   else
   return false;
}

bool isOperator(char ch)
{
   if(ch == '=' || ch == '+' || ch == '-' || ch == '*' || ch == '/' || ch == '%')
   return true;
   else
   return false;
}
bool isValidIdentifier(string str)
{
   if(str[0] == '0' || str[0] == '1' || str[0] == '2' || str[0] == '3' || str[0] == '4' || str[0] == '5' ||str[0]
== '6' || str[0] == '7' || str[0] == '8' || str[0] == '9' || isDelimiter(str[0]) == true)
   return false;
   else
   return true;
}
bool isKeyword(string str)
{
   if(str == "int" || str == "char" || str == "void" || str == "for" || str == "while")
   return true;
   else
   return false;
}
bool isInteger(string str)
{
   int size = str.size();
   if(size == 0)
   return false;
   for(int i =0; i<size; ++i)
   {
      if(str[i] != '0' || str[i] != '1' || str[i] != '2' || str[i] != '3' || str[i] != '4' || str[i] != '5' || str[i] !=
'6' || str[i] != '7' || str[i] != '8' || str[i] != '9')
      return false;
   }
   return true;
}
string subString(string str ,int left, int right)
{
    string s;
    for(int i=left; i<=right; ++i)
    {
        s.push_back(str[i]);
    }
    return s;
}


void parse(string str)
{
    int left = 0, right = 0;
    int len = str.size();

    while (right <= len && left <= right)
            {
        if (isDelimiter(str[right]) == false)
            right++;

        if (isDelimiter(str[right]) == true && left == right)
                        {
            if (isOperator(str[right]) == true)
                cout<<str[right]<<" IS AN OPERATOR\n";

            right++;
            left = right;
        }
                        else if (isDelimiter(str[right]) == true && left != right || (right == len && left != right))
                        {
            string subStr = subString(str, left, right - 1);

            if (isKeyword(subStr) == true)
                cout<<subStr<<" IS A KEYWORD\n";

            else if (isInteger(subStr) == true)
                cout<<subStr<<" IS AN INTEGER\n";

            else if (isValidIdentifier(subStr) == true && isDelimiter(str[right - 1]) == false)
               cout<<subStr<<" IS A VALID IDENTIFIER\n";


            left = right;
        }
    }
    return;
}
```

```
int main()
{
    string str = "int a = b + c";
    parse(str);
    return 0;
}
```

Practical – 2

Objective- Write a program to find comment in an expression.

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

void isComment(string line)
{
    if (line.size()>=2 && line[0] == '/' && line[1] == '/')
    {
        cout << "It is a single-line comment";
        return;
    }

    if (line.size()>=4 && line[0] == '/' && line[1] == '*' && line[line.size() - 1] == '/' &&
line[line.size() - 2] == '*')
    {
        cout << "It is a multi-line comment";
        return;
    }
    else
    cout << "It is not a comment";
}
int main()
{
    string line = "/*Finding comment in an expression*/";
    cout<<line1<<endl<<endl;
    isComment(line);
    return 0;
}
```

Practical – 3

Objective- Write a program to find if statement in an expression.

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

bool check( string input)
{
    int if_pos = -1;
    int else_pos = -1;
    for (int i = 0; i < input.size() - 1; i++)
    {
        if (input[i] == 'i' && input[i+1] == 'f')
            if_pos = i;
        if (input[i] == 'e' && input[i+1] == 'l' && input[i+2] == 's' && input[i+3] == 'e')
            else_pos = i;
    }

    if(if_pos != -1 && else_pos != -1 && else_pos > if_pos)
    return true;
    else
    return false;
}

int main()
{
    string input = "if (x > 0) { y = x; } else { y = 0; }";
    cout<<"INPUT: "<<input<<endl<<endl;
    bool is_if_else = check(input);
    if (is_if_else)
        cout << "The input contains an if-else statement." <<endl;
    else
        cout << "The input does not contain an if-else statement." <<endl;

    return 0;
}
```

Practical – 4

Objective- Write a program to find tokens in an arithmetic expression.

```
#include <bits/stdc++.h>
using namespace std;
bool isDigit(char s)
{
        int ch=s;
        if(ch == '0' || ch == '1' || ch == '2' || ch == '3' || ch == '4' || ch == '5' || ch == '6' || ch == '7' ||
ch == '8' || ch == '9')
        {
            return true;
        }
    return false;
}
bool isOperator(char e)
{
    if(e == '+' || e == '-' || e == '*' || e == '/' || e == '^' || e == '%' || e == '=' )
        return true;
    else
        return false;
}
bool isParentheses(char s)
{
    if(s == '(' || s== ')' )
        return true;
    else
        return false;
}

int main()
{
    string expression = "5*(9+4)/int32_t";
    for(int i=0; i<expression.size(); i++)
    {
        char ch=expression[i];
        if(isDigit(ch))
        {
            cout<<ch<<" is a number"<<endl;
        }
        else if(isOperator(ch))
        {
            cout<<ch<<" is an operator "<<endl;
        }
        else if(isParentheses(ch))
{
```

```
          cout<<ch<<" parentheses"<<endl;
        }
    }
    return 0;
}
```