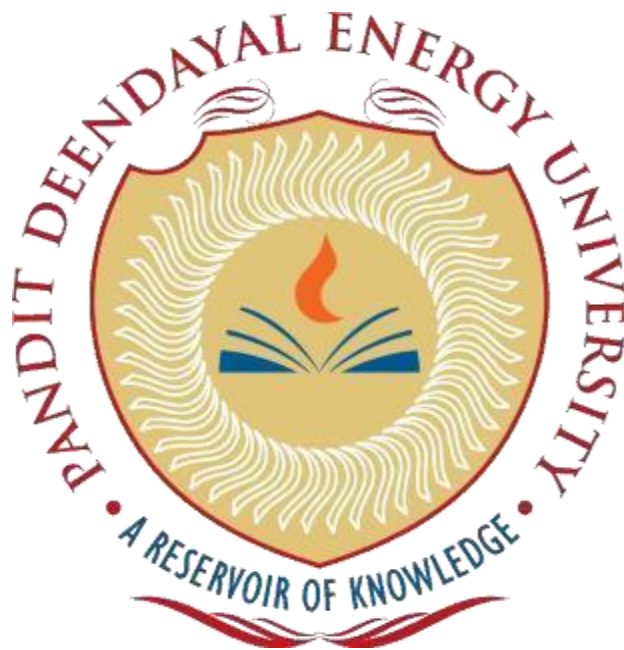


Pandit Deendayal Energy University, Gandhinagar

School of Technology

Department of Computer Science & Engineering

System Software & Compiler Design Lab (20CP302P)



Name: Sutariya Smit Dharmendrabhai

Enrolment No: 21BCP142

Semester: V

Division: 3 (G5)

Branch: Computer Science Engineering

Practical: 4

Aim: WAP to implement Recursive Decent Parser (RDP) parser for given grammar.

Code:

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

int ptr;
string input;
bool E();
bool E_();
bool T();
bool T_();
bool F();
bool F_();
bool P();

/*
E  -> T E'
E' -> + T E' | - T E' | (empty)
T  -> F T'
T' -> * F T' | / F T' | (empty)
F  -> P F'
F' -> ^ F | (empty)
P  -> ( E ) | i
*/

bool E()
{
    int fallback = ptr;
    if(T()){
        if(E_()){
            return true;
        }
    }
    ptr = fallback;
    return false;
}

bool E_()
{

```

```
int fallback = ptr;
if(ptr < input.length() && (input[ptr] == '+' || input[ptr] == '-')){
    ptr++;
    if(T()){
        if(E_()){
            return true; // t means that + or - is selected
        }
    }
    ptr = fallback;
    return false;
}
return true; // it means that epsilon is selected
}

bool T()
{
    int fallback = ptr;
    if(F()){
        if(T_()){
            return true;
        }
    }
    ptr = fallback;
    return false;
}

bool T_()
{
    int fallback = ptr;
    if(ptr < input.length() && (input[ptr] == '*' || input[ptr] == '/')){
        ptr++;
        if(F()){
            if(T_()){
                return true;
            }
        }
        ptr = fallback;
        return false;
    }
    return true; // it means that epsilon is selected
}

bool F()
{
    int fallback = ptr;
```

```
    if(P()){
        if(F_()){
            return true;
        }
    }
    ptr = fallback;
    return false;
}

bool F_()
{
    int fallback = ptr;
    if(ptr < input.length() && input[ptr] == '^'){
        ptr++;
        if(F()){
            return true;
        }
        ptr = fallback;
        return false;
    }
    return true;
}

bool P()
{
    int fallback = ptr;
    if(ptr < input.length() && input[ptr] == '('){
        ptr++;
        if(E()){
            if(ptr < input.length() && input[ptr] == ')'){
                ptr++;
                return true;
            }
        }
        ptr = fallback;
        return false;
    }
    else if(ptr < input.length() && input[ptr] == 'i') {
        ptr++;
        return true;
    }
    return false;
}

int main()
```

```
{
    cout << "Enter the input string: ";
    getline(cin, input);

    if(input.length() < 1)
    {
        cout << "The input string is invalid!!" << endl;
    }

    ptr = 0;
    bool isValid = E();
    if(isValid && ptr == input.length())
    {
        cout << "The input string is valid." << endl;
    }
    else
    {
        cout << "The input string is invalid!" << endl;
    }
}
```

Output:

```
PS D:\Sem-5\compiler\Lab4> cd "d:\Sem-5\compiler\Lab4\" ;
Enter the input string: i+i*i(i-i/i)^i
The input string is invalid!
PS D:\Sem-5\compiler\Lab4> cd "d:\Sem-5\compiler\Lab4\" ;
Enter the input string: i+i*i+(i-i/i)^i
The input string is valid.
PS D:\Sem-5\compiler\Lab4> █
```