## 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:**

