Practical No: 2

Name : Patel Savankumar P.
Enroll No : 19BCE519
Subject : Compiler Construction

AIM: To implement a Recursive Descent Parser Algorithm for the grammar.

Code:

```
// Valid : i$, i+i$, i-i$, i+i-i$

// Grammar:
//
// E --> iS
// S --> +iS | -iS | e

#include<stdio.h>

char l;

void E();
void S();

int ok = 1;

void match(char t)
{
    if(t==l)
    {
        l = getchar();
    }
    else
    {
```

```
        ok = 0;
    }
}

void E()
{
    match('i');
    S();
}

void S()
{
    if(l=='+')
    {
        match('+');
        match('i');
        S();
    }
    else if(l=='-')
    {
        match('-');
        match('i');
        S();
    }
    else
    {
        return;
    }
}

int main()
{
    l = getchar();
    E();

    if(l!='$')
    {
        ok = 0;
    }
```

```
    printf(ok?"String is Passed ":"String is not Passed");

    return 0;
}
```

Output:

```
PS E:\Semester 7\CC\Lab> gcc .\Prac2.c
PS E:\Semester 7\CC\Lab> ./a.exe
i+i-i$
String is Passed
PS E:\Semester 7\CC\Lab> ./a.exe
ii+i$
String is not Passed
PS E:\Semester 7\CC\Lab> ./a.exe
i-i-i+i-i$
String is Passed
PS E:\Semester 7\CC\Lab> ./a.exe
i-i-i+i-i-$
String is not Passed
PS E:\Semester 7\CC\Lab>
```

Conclusion:

From this practical I learned how to create a recursive descent parser to accept or deny any string for any grammer.