**Practical 2**

**Compiler Construction**

2CS701

**Mistry Unnat**

20BCE515



Department of Computer Science and Engineering

Institute of Technology

Nirma University

Ahmedabad

**Aim :**

**To implement a** **Recursive Descent Parser Algorithm for**

**the grammar.**

**Production Rules :**

S -> aSa | T

T -> bT | ε

**Grammar :**

Null strings or strings containing any number on ‘b’ or strings starting and ending with ‘a’ and having any number of ‘b’ between them.

**C code :**

#include <stdio.h>

#include <string.h>

char input[100];

int i = 0;

int T()

{

    if (input[i] == 'b')

    {

        i++;

        if (T())

            return 1;

        else

            return 0;

    }

    else

        return 1;

}

int S()

{

    if (input[i] == 'a')

    {

        i++;

        if (S())

        {

            if (input[i] == 'a')

            {

                i++;

                return 1;

            }

            else

                return 0;

        }

        else

            return 0;

    }

    else if (T())

    {

        return 1;

    }

    else

        return 0;

}

void main()

{

    printf("Enter the string to be checked : ");

    gets(input);

    if (S())

    {

        if (input[i] == '\0')

            printf("String is accepted\n");

        else

            printf("String is not accepted\n");

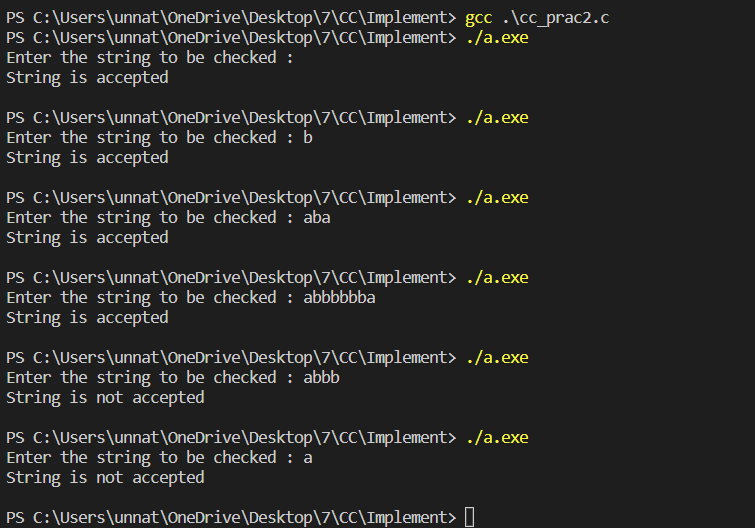
    }

    else

        printf("String is not accepted\n");

}

**Output :**

****

**Conclusion :**

From this practical, we learnt about Recursive Descent Parser Algorithm.