Name: Gokulnath M Prabhu
Class: CS7B
Roll No: 21

# Lab Cycle 3 - Experiment 15

Design and implement a recursive descent parser for a given grammar

**Code:**

```c
#include <stdio.h>
char inp[100];
int len = 0;
int curr = 0;
int E();
int Z();
int main()
{
   printf("Enter input:\n");
   scanf("%s", inp);
   while (inp[len] != '\0')
       len++;
   int res = E();
   if (res == 1 && curr == len)
       printf("Input has been accepted.\n");
   else
       printf("Input has been rejected.\n");
}

int E()
{
   int result;
   if (inp[curr] == 'i')
   {
       curr++;
       result = Z();
       if (result == 1)
           return 1;
       else
           return -1;
   }
   return -1;
}

int Z()
{
   int result;
```

```
    if (inp[curr] == '+' && inp[curr + 1] == 'i')
    {
        curr += 2;
        result = Z();
        if (result == 1)
            return 1;
    }
    return 1;
}
```

**Output:**

```
● → Recursive_Descent_Parser git:(master) ✗ gcc recursive_descent.c
● → Recursive_Descent_Parser git:(master) ✗ ./a.out
  Enter input:
  i+j+k
  Input has been rejected.
● → Recursive_Descent_Parser git:(master) ✗ ./a.out
  Enter input:
  i=j=k
  Input has been rejected.
○ → Recursive_Descent_Parser git:(master) ✗ _
```