COMP 451

FORMAN CHRISTIAN COLLEGE

(A CHARTERED UNIVERSITY)

COMPILER CONSTRUCTION

LAB 7

ROLL No. **Time Allowed: 120 min**

This is an online lab. No group formation is allowed. It's an open books and open notes lab session. You CANNOT share your code with your class fellows.

Grading Criteria

Working Code in class: 70%

Properly formatted Report: 30%

Lab Task 1 [20 Marks]

Go through the following code. This program implements the working of a recursive decent parser for the grammar shown below:

```
      E --> iE'
      E' --> +iE' | eps
```

```
#include <stdio.h>
#include <string.h>
int E();
int E_prime();
char expr[100];
int count,l;

int main()
{
      count = 0;
      printf("\nRecursive descent parsing for the following grammar\n");
      printf("\nE->iE'\nE'->+iE'| @\n");
      printf("\nEnter the string to be checked:");

      fgets(expr,100,stdin);
      if(E())
      {
            if(expr[count]=='$')
                  printf("\nString is accepted");
            else

                  printf("\nString is not accepted");

      }

      else{
            printf("\nString not accepted");
      }
      return 0;

}
```

```
int E()
{
      //E --> iE'
      if(expr[count] == 'i')
      {
            count++;
            if(E_prime())
            {
                  return 1;
            }
            else
                  return 0;
      }
      else
            return 0;

}

int E_prime()
{
      //E' --> +iE' | e
      if(expr[count] == '+')
      {
            count++;
            if(expr[count] == 'i'){
                  count++;
                  if(E_prime())
                        return 1;
                  else
                        return 0;
            }
            else
                  return 0;

      }
      else
            return 1;


}
```

Type this code on your console. Run it and provide different input strings. Some valid input strings for the given grammar are:

- `i$`
- `i+i+i$`
- `i+i+i+i+i$`

Some invalid strings are:

- `ii$`
- `i++$`

Lab Task 2 [50 Marks]

Once you have successfully run and understood the working of the above program, now you need to write a program on same lines, that should implement the recursive decent parser algorithm for the following grammar:

```
S → rXd | rZd
X → oa | ea
Z → ai
```

Input strings that this parser accepts are: read, road, raid. Make sure your input string must ends with a ‘$’ symbol.

Important Note

Submit your work on Moodle course page within the time specified on Moodle. Any late submission will be graded with a cap of 50%.

Your report should carry appropriate output screenshots for both the above programs.