LAB 9 - BOTTOM PARSER FOR SIMPLE GRAMMAR

Name: Pranamya G Kulal
Class: CSE A
Roll no: 8
Reg no: 220905018

Q1) 1. Develop an SLR(1) parser for the given expression grammar and demonstrate parsing actions.
E->E+T|T
T-> T*F|F
F-> (E)|id

i)Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char input[100];
char a[3];
char* symbol_col[6] = {"id","+","*","(",")","$"};
int i = 0;
int production_len[7] = {1,3,1,3,1,3,1};
int prod_head[7] = {0,1,1,2,2,3,3}; // corresponding col id for prodn heads in Goto
char * prod[7] = {"E'->E","E->E+T","E->T","T->T*F","T->F","F->(E)","F->id"};
char * Action[12][6] ={
   {"s5","","","s4","",""},
   {"","s6","","","","accept"},
   {"","r2","s7","","r2","r2"},
   {"","r4","r4","","r4","r4"},
   {"s5","","","s4","",""},
   {"","r6","r6","","r6","r6"},
   {"s5","","","s4","",""},
   {"s5","","","s4","",""},
   {"","s6","","","s11",""},
   {"","r1","s7","","r1","r1"},
   {"","r3","r3","","r3","r3"},
   {"","r5","r5","","r5","r5"}
};

int Goto[12][4] ={
   {-1,1,2,3},
   {-1,-1,-1,-1},
   {-1,-1,-1,-1},
   {-1,-1,-1,-1},
   {-1,8,2,3},
   {-1,-1,-1,-1},
   {-1,-1,9,3},
   {-1,-1,-1,10},
   {-1,-1,-1,-1},
   {-1,-1,-1,-1},
   {-1,-1,-1,-1},
```

```
    {-1,-1,-1,-1}
};

int getnextsymbol(){
    a[0] = input[i++];
    if(a[0] == 'i' && input[i]== 'd'){
        a[1] = input[i++];
        a[2] = '\0';
    }
    else a[1] = '\0';
    for(int j=0; j<6;j++)
        if(strcmp(a,symbol_col[j])==0) return j;
    printf("Incorrect symbol\n");
    exit(-1);
}
int main(){
    int stack[50];
    int stackptr = -1;
    printf("Enter input: ");
    scanf("%s",input);
    stack[++stackptr]= 0;
    int col = getnextsymbol();
    while(1){
        int s = stack[stackptr];
        if(Action[s][col][0] == 's'){
            stack[++stackptr] = Action[s][col][1] - '0';
            col = getnextsymbol();
        }
        else if(Action[s][col][0] == 'r'){
            int prodn = Action[s][col][1] - '0';
            int prodlen = production_len[prodn];
            for(int k = 0; k < prodlen; k++) stackptr--;
            int t = stack[stackptr];
            stack[++stackptr] = Goto[t][prod_head[prodn]];
            printf("%s %s\n",Action[s][col],prod[prodn]);
        }
        else if(strcmp(Action[s][col],"accept")==0){
            printf("Success\n");
            break;
        }
        else{
            printf("Error\n");
            exit(-1);
        }
    }
}
```

ii) Output

```
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab9-BottomParserForSimpleGrammar$ gcc lab9q1.c -o lab9q1
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab9-BottomParserForSimpleGrammar$ ./lab9q1
```

Enter input: id+id*id$
r6 F->id
r4 T->F
r2 E->T
r6 F->id
r4 T->F
r6 F->id
r3 T->T*F
r1 E->E+T
Success