Practical-6

Implement Recursive Descendent Parsing for the given Grammar.

E -> T + E / T

T -> F * T / F

F -> (E) / i

```
#include "stdio.h"
#include "conio.h"
char input[100];
char prod[100][100];
int pos=-1,l,st=-1;
char id,num;
void E();
void T();
void F();
void advance();
void Td();
void Ed();
void advance()
{
	pos++;
	if(pos<l)
	{
		if(input[pos]>='0'&& input[pos]<='9')
		{
			num=input[pos];
			id='\0';
		}
		if((input[pos]>='a' || input[pos]>='A')&&(input[pos]<='z' || input[pos]<='Z'))
		{	id=input[pos];
```

```
                        num='\0';
                }
        }
}
void E()
{
        strcpy(prod[++st],"E->TE'");
        T();
        Ed();
}
void Ed()
{
        int p=1;
        if(input[pos]=='+')
        {
                p=0;
                strcpy(prod[++st],"E'->+TE'");
                advance();
                T();
                Ed();
        }
        if(input[pos]=='-')
        {
                p=0;
                strcpy(prod[++st],"E'->-TE'");
                advance();
                T();
                Ed();
        }
        if(p==1)
                strcpy(prod[++st],"E'->null");
```

```
}
void T()
{
	strcpy(prod[++st],"T->FT'");
	F();
	Td();
}
void Td()
{
	int p=1;
	if(input[pos]=='*')
	{
		p=0;
		strcpy(prod[++st],"T'->*FT'");
		advance();
		F();
		Td();
	}
	if(input[pos]=='/')
	{
		p=0;
		strcpy(prod[++st],"T'->/FT'");
		advance();
		F();
		Td();
	}
	if(p==1)
	strcpy(prod[++st],"T'->null");
}
void F()
{
```

```
        if(input[pos]==id)
        {
                strcpy(prod[++st],"F->id");
                advance();
        }
        if(input[pos]=='(')
        {
                strcpy(prod[++st],"F->(E)");
                advance();
                E();
                if(input[pos]==')')
                {
                        //strcpy(prod[++st],"F->(E)");
                        advance();
                }
        }
        if(input[pos]==num)
        {
                strcpy(prod[++st],"F->num");
                advance();
        }
}
int main()
{
        int i;
        printf("Enter Input String ");
        scanf("%s",input);
        l=strlen(input);
        input[l]='$';
        advance();
        E();
```

```
        if(pos==l)
        {
                printf("String Accepted\n");
                for(i=0;i<=st;i++)
                        printf("%s\n",prod[i]);
        }
        else
                printf("String rejected\n");
        getch();
        return 0;
}
```

OUTPUT:

```
Enter Input String (a+b)*c
String Accepted
E->T+E
T->F*T
F->(E)
E->T
T->F
F->i
E->T
```