Experiment No 3

Aim : Design recursive descent parser

Source Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define SIZE 100
int i = 0;
// T'->*FT'/eps
void procTdash(char str[])
{
    if(str[i]=='*'){
        i++;
        procF(str);
        procTdash(str);    }}
// F->(E)/i
void procF(char str[])
{
    if(str[i]=='('){
        i++;
        procE(str);
        if(str[i]==')'){
            i++;
        }
        else{
            printf("ERROR\n");
        }
    }
    else if(str[i]=='i'){
        i++;
    }
    else{
        printf("ERROR\n");
    }
}
// T->FT'
void procT(char str[])
{
    procF(str);
    procTdash(str);
}
// E'->+TE'/eps
void procEdash(char str[])
{
    if(str[i]=='+'){
        i++;
        procT(str);
        procEdash(str);    }}
//E->TE'
void procE(char str[])
{
    procT(str);
    procEdash(str);}
int main(){
    printf("Enter the input: \n");
    char str[SIZE];
    fgets(str, SIZE, stdin);
    int n = strlen(str);
    procE(str);
    if((n-1)==i)    {
        printf("Input is
accepted\n");    }
    else{
        printf("Input isn't
accepted\n");
        printf("ERROR");    }
    return 0;}
```

Output:

```
Enter the input:
i+i
Input is accepted
```

```
Enter the input:
i+
ERROR
```

```
Enter the input:
(i+i*i
ERROR
```

```
Enter the input:
(i+i*i)
Input is accepted
```

```
Enter the input:
i/i
Input isn't accepted
ERROR
```