Exp No: 4

IMPLEMENTATION OF RECURSIVE DESCENT PARSER IN C

Date: 23/02/2021

Name: MADHUMITHA S

Register Number: 185001086

---

1) **Aim:**

To implement Recursive Descent parser using C.

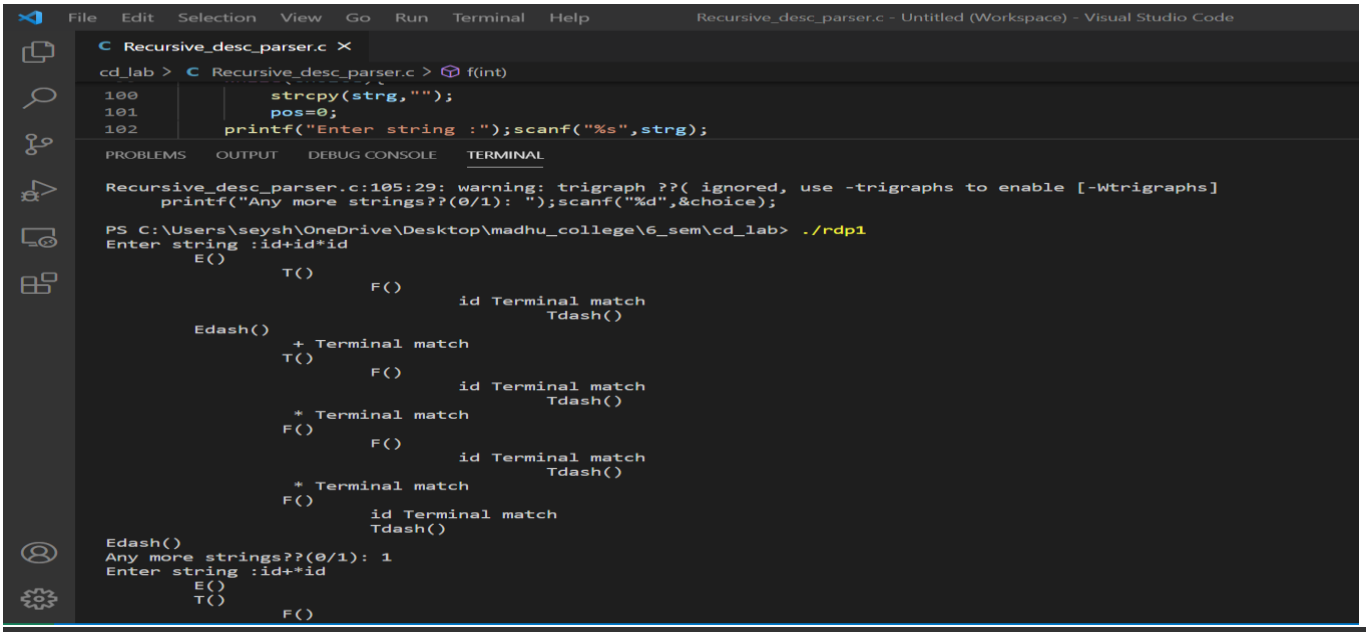
2) PROGRAM CODE:

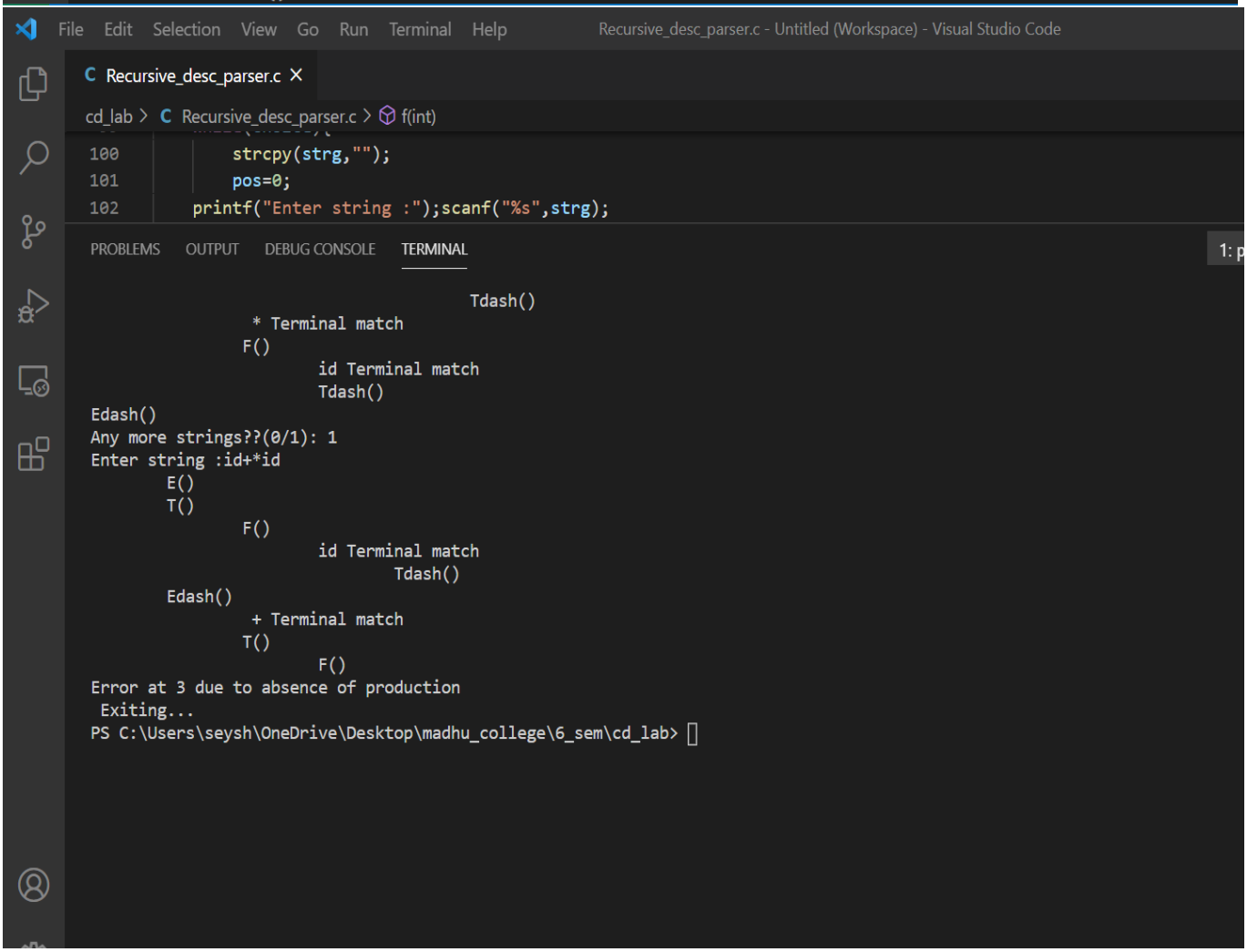
```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void e(int s);
void edash(int s);
void f(int s);
void t(int s);
void tdash(int s);
void match(char c,int s);
char strg[20];
int pos=0;
int space=1;


void e(int s){
    int i=0;
    for (i=0;i<s;i++){printf("\t");}
    printf("E()\n");space++;
    t(space);space++;
    edash(space);
    space=0;
}
void t(int s){
    int i=0;
    for (i=0;i<s;i++){printf("\t");}
    printf("T()\n");space++;
    f(space);space++;
    tdash(space);
    space=0;
}
void match(char c,int s){
    int i=0;
    if(strg[pos]==c){
```

```
    for (i=0;i<s;i++){printf("\t");}
    if(c=='i'){printf("id Terminal match\n");}
    if(c!='d' && c!='i'){printf(" %c Terminal match\n",c);}
        pos++;
    }
    else{
        printf("Error at %d due to mismatch \n Exiting...",pos);
        exit(0);
    }
}

void edash(int s){
    int i=0;
    for (i=0;i<s;i++){printf("\t");}
    printf("Edash()\n");
    if(strg[pos]=='+'){
        space++;
        match('+',space);
        t(space);
        edash(space);
        space=0;

    }
}

void tdash(int s){
    int i=0;
    for (i=0;i<s;i++){printf("\t");}
    printf("Tdash()\n");
    if(strg[pos]=='*'){
        space++;
        match('*',space);
        f(space);
        tdash(space);
        space=0;
    }

}
void f(int s){
   int i=0;
    for (i=0;i<s;i++){printf("\t");}
     printf("F()\n");
    if(strg[pos]=='i'){
        space++;
        match('i',space);
        match('d',space);
        space=0;
```

```
    }
    else if(strg[pos]=='('){
        space++;
        match('(',space);space++;
        e(space);space++;
        match(')',space);

    }
    else{
        printf("Error at %d due to absence of production\n Exiting...",pos);
        exit(0);
    }
}
int main(){
    int choice=1;
    while(choice){
        strcpy(strg,"");
        pos=0;
    printf("Enter string :");scanf("%s",strg);
    e(1);

    printf("Any more strings??(0/1): ");scanf("%d",&choice);
    }
    return 0;
}
```

3)OUTPUT SCREENSHOTS:





4)LEARNING OUTCOME:

- Implementing a recursive descent parser for the given grammar.

- Parsing a given string using the RD parser.
- Checking the sequence of procedure calls when a string is being parsed or throwing and error if not able to parse.