10. Write a C program to construct recursive descent parsing.

#include <stdio.h>

#include <string.h>

#define SUCCESS 1

#define FAILED 0

int E(), Edash(), T(), Tdash(), F();

const char \*cursor;

char string[64];

int main()

{

    puts("Enter the string");

    // scanf("%s", string);

    sscanf("i+(i+i)\*i", "%s", string);

    cursor = string;

    puts("");

    puts("Input      Action");

    puts("--------------------------------");

    if (E() && \*cursor == '\0') {

        puts("--------------------------------");

        puts("String is successfully parsed");

        return 0;

    } else {

        puts("--------------------------------");

        puts("Error in parsing String");

        return 1;

    }

}

int E()

{

    printf("%-16s E -> T E'\n", cursor);

    if (T()) {

        if (Edash())

            return SUCCESS;

        else

            return FAILED;

    } else

        return FAILED;

}

int Edash()

{

    if (\*cursor == '+') {

        printf("%-16s E' -> + T E'\n", cursor);

        cursor++;

        if (T())

        {

            if (Edash())

                return SUCCESS;

            else

                return FAILED;

        }

      else

            return FAILED;

    }

  else

    {

        printf("%-16s E' -> $\n", cursor);

        return SUCCESS;

    }

}

int T()

{

    printf("%-16s T -> F T'\n", cursor);

    if (F())

    {

        if (Tdash())

            return SUCCESS;

        else

            return FAILED;

    } else

        return FAILED;

}

int Tdash()

{

    if (\*cursor == '\*')

    {

        printf("%-16s T' -> \* F T'\n", cursor);

        cursor++;

        if (F())

        {

            if (Tdash())

                return SUCCESS;

            else

                return FAILED;

        } else

            return FAILED;

    }

  else

    {

        printf("%-16s T' -> $\n", cursor);

        return SUCCESS;

    }

}

int F()

{

    if (\*cursor == '(')

    {

        printf("%-16s F -> ( E )\n", cursor);

        cursor++;

        if (E())

        {

            if (\*cursor == ')')

            {

                cursor++;

                return SUCCESS;

            } else

                return FAILED;

        } else

            return FAILED;

    } else if (\*cursor == 'i')

    {

        cursor++;

        printf("%-16s F ->i\n", cursor);

        return SUCCESS;

    } else

        return FAILED;

}  