DATE 22-12-2021

EXPERIMENT 10

AIM

Constonct a recursive descent parser for an experession.

PROGRAM

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
// Decloration
void Tprime();
 :

int main() {
  count = 0;
  flag = 0;
  printf("Enter an Algebraic Expression");
  scanf("%s", expression);
  E();
  if ((strlen(expression) == count) && (flag == 0)) {
        printf("The Expression %s is valid \n, expression);
  else
        printf("\n The expression %s is Invalid \n", expression);

}
void E() {
   T()
   Eprime();

}
```

```
void T() {
     check();
     Tprime();

}
```

```
Tprime() {
   if (expression [count] == '*') {
        Count ++;
        check();
        Tprime();
   }
}
```

```
void check () {
   if (isalnum(expression [comt]))
          comt++;
   else if (expression [comt] == '(') {
          comt ++;  E(); if exp( == ')'); else flag = 1;
   } else
          flag = 1;
```

```
void E prime() {
    if (expression [count] == '+') {
            Count ++;
            T();
            Eprime();
    }
}
```

OUTPUT

```
Enter an Algebraic Expression: i+ i*i
 The Expression i+ i*i is valid

Enter an Algebraic Expression i + (i+i)
 The expression i + (i*i) is valid.
```