Practical 10

Aim: Write program to implement Recursive Descent Parser for the given grammar.

E -> T + E | T

T -> F * T | F

F -> id

Code:

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
char input[10];
int i,error;
void E();
void T();
void Eprime();
void Tprime();
void F();
int main(){
                i=0;
                error=0;
        printf("Enter an arithmetic expression   :  ");
        gets(input);
        E();
        if(strlen(input)==i&&error==0)
             printf("\nAccepted…\n");
        else printf("\nRejected…\n");
        return 0;
                }
void E()
{
   T();
   Eprime();
}
void Eprime()
{
   if(input[i]=='+')
   {
   i++;
   T();
   Eprime();
   }
   }
void T()
{
   F();
   Tprime();
}
void Tprime()
{
   if(input[i]=='*')
```

```
    {
            i++;
            F();
            Tprime();
            }
}
void F()
    {
        if(isalnum(input[i]))i++;
        else if(input[i]=='(')
        {
        i++;
        E();
        if(input[i]==')')
        i++;
        else error=1;
         }
        else error=1;
}
```

Output:

Conclusion:

We learned to implement and validate the RDP for the given grammar in this practical.