

Practical-9

AIM : Write a program to implement Recursive Decent Parser for following grammar and check given input strings accepted by grammar or not?

expr \rightarrow digit rest

rest \rightarrow +digit rest | - digit rest | ϵ

digit \rightarrow 0 | 1 | 2 | 3 | ... | 9

❖ CODE :

```
#include<stdio.h>
#include<ctype.h>
#include<string.h>
void Tprime();
void Eprime();
void E();
void check();
void T();
void dollar();
char expression[10];
int count, flag;
int main()
{
    count = 0;
    flag = 0;

    printf("\nEnter an Algebraic Expression:\t");
    scanf("%s", expression);
    E();

    if((strlen(expression) == count) && (flag == 0))
    {
        printf("\nThe Expression %s is Valid\n", expression);
    }
    else
    {
        printf("\nThe Expression %s is Invalid\n", expression);
    }
}
```

```
    }

}

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

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

void Tprime()
{
    if(expression[count] == '-')
    {
        count++;
        check();
        Tprime();
    }

}

void check()
{
    if(isalnum(expression[count]))
    {
        count++;
    }
    else
    {
        flag = 1;
    }

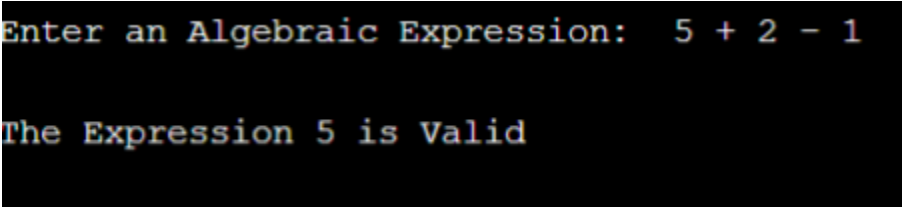
}

}

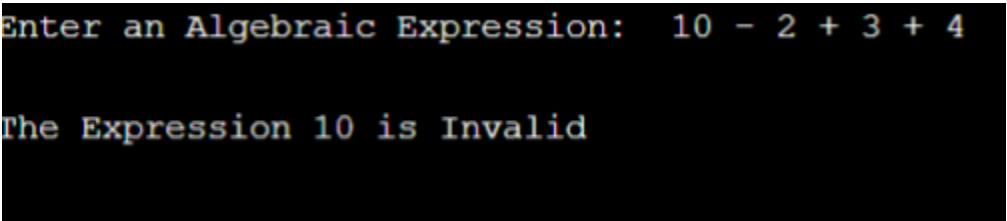
void Eprime()
{
    if(expression[count] == '+')
    {
        count++;
        T();
        Eprime();
    }
}
```

```
}  
}  
void dollar()  
{  
    if(expression[count] == '$') count++;  
}
```

❖ OUTPUT



Enter an Algebraic Expression: 5 + 2 - 1
The Expression 5 is Valid



Enter an Algebraic Expression: 10 - 2 + 3 + 4
The Expression 10 is Invalid