1. 과제에 대한 설명

본 프로그램은 재귀하강형 파싱 기술을 이해하여 입력되는 수식을 정수, 덧셈, 곱셈, 괄호의 문법 검사하고 값을 계산하여 그 결과를 출력하는 c 언어 프로그램이다.

수식을 검사하는 과정에서 main() 함수는 expression() 함수를 호출하고, expression() 함수는 term() 함수를 호출한다. term() 함수는 factor() 함수를 호출하고, factor() 함수는 토큰 종류를 분석하여 재귀적으로 다시 expression() 함수를 호출할 수 있다.

get_token() 함수가 호출되면, 입력된 수식을 어휘 분석하여 토큰으로 자르고, 해당 토큰이 {'0'~'9', '+', '*', '(', ')', 'EOF', 기타} 중 어떤 종류에 속하는지 판별한 후, 전역 변수 token의 값을 저장한다. 그리고 그 토큰이 정수인 경우, 정수의 값을 전역변수 num에 저장하도록 구성하였다. 또한, 토큰의 값이 숫자일 경우, 여러 자리 정수형 숫자를 인식할 수 있는 기능을 추가하였다.

expression() 함수는 term들의 더하기로 이루어진 expression 값을 계산하고, term() 함수는 factor들의 곱하기로 이루어진 term 값을 계산한다. factor() 함수는 해당 토큰이 정수일 경우, 전역 변수 num 값으로부터 계산하고, 괄호 사이에 expression이 나오는 경우 괄호 사이의 수식의 값을 재귀적으로 계산하도록 expression() 함수를 호출하여 계산한다. main() 함수는 마지막으로 expression 뒤에 아무것도 없는지 확인하는 방식으로 프로그램이 동작된다.

2. 프로그램 소스코드

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 64

int num;
int arrNum = 0;
char ch = ' ';
char arr[MAX];
typedef enum TOKEN {null, NUMBER, PLUS, STAR, LP, RP, END} T;
T token;

void get_token();
int expression();
int term();
int factor();
void error(int i);

void get_token(){
        ch = arr[arrNum];
        if(ch == '+'){
                token = PLUS;
        }
        else if(ch == '*'){
                token = STAR;
        }
        else if(ch == '('){
                token = LP;
        }
        else if(ch == ')'){
                token = RP;
        }
        else if(ch == '\n'){
                token = END;
        }
        else if((ch >= '0') && (ch <= '9')){
```

```
                token = NUMBER;
                num = ch - '0';
                while((arr[arrNum+1] >= '0') && (arr[arrNum+1] <= '9'))
                {
                        arrNum++;
                        ch = arr[arrNum];
                        num = num*10 + (ch - '0');
                }
        }
        else{
                token = null;
        }
        arrNum++;
}

int expression(){
        int result;
        result = term();
        while(token==PLUS)
        {
                get_token();
                result=result+term();
        }
        return (result);
}int term(){
        int result;
        result=factor();
        while(token==STAR)
        {
                get_token();
                result=result*factor();
        }
        return (result);
}

int factor(){
        int result;
```

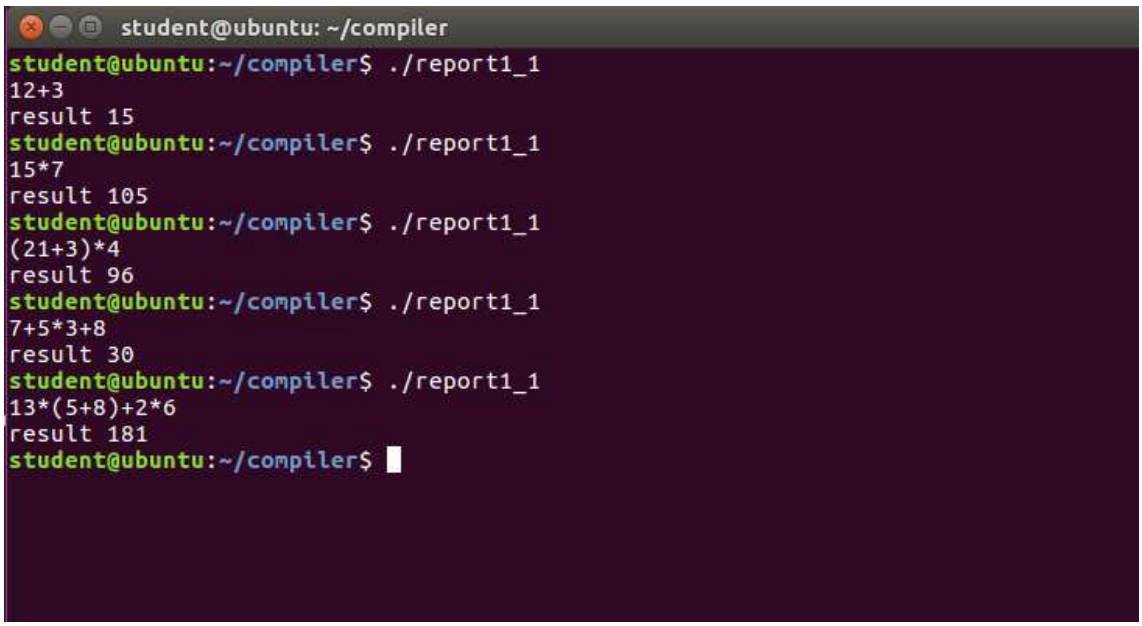
```
        if(token == NUMBER)
        {
                result=num;
                get_token();
        }
        else if(token==LP){
                get_token();
                result=expression();
                if(token == RP)
                        get_token();
                else
                        error(2);
        }
        else
                error(1);
        return (result);
}

void main(){
        int result;
        int i = 0;
        while(i<MAX){
                arr[i] = getchar();
                if(arr[i] == '\n')
                        break;
                i++;
        }
        get_token();
        result=expression();
        if(token!=END){
                error(3);
        }
        else
                printf("result %d \n",result);
}

void error(int i){
```

```
        switch(i)
        {
                case 1:
                        printf("error : number or '(' expected\n");
                        break;
                case 2:
                        printf("error : ')' expected\n");
                        break;
                case 3:
                        printf("error : EOF expected\n");
                        break;
        }
        exit(1);
}
```

3. 실행 결과


student@ubuntu: ~/compiler
student@ubuntu:~/compiler$ ./report1_1
12+3
result 15
student@ubuntu:~/compiler$ ./report1_1
15*7
result 105
student@ubuntu:~/compiler$ ./report1_1
(21+3)*4
result 96
student@ubuntu:~/compiler$ ./report1_1
7+5*3+8
result 30
student@ubuntu:~/compiler$ ./report1_1
13*(5+8)+2*6
result 181
student@ubuntu:~/compiler$