

[~] 구현 처리 하였습니다.

C++

```

1  #include <ctype.h>
2  #include <iostream>
3  #include <stdlib.h>
4
5  using namespace std;
6
7  char token;
8
9  void command(void);
10 int expr(void);
11 int term(void);
12 int factor(void);
13 int number(void);
14 int digit(void);
15
16 void error(void) {
17     cout << "syntax error!" << endl;
18     exit(1);
19 }
20
21 void getToken(void) {
22     while (true) {
23         token = getchar();
24         if (token != ' ')
25             return;
26     }
27 }
28
29 void match(char c) {
30     if (token == c)
31         getToken();
32     else
33         error();
34 }
35
36 void command(void) {
37     int result = expr();
38     if (token == '\n') {
39         cout << result << endl;
40     } else {
41         error();
42     }
43 }
44
45 // <number> -> <digit><digit>
46 int number(void) {
47     int result = digit();
48     while (isdigit(token))
49         result = 10 * result + digit();
50     return result;
51 }
52
53 // digit -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
54 int digit(void) {
55     int result;
56     if (isdigit(token)) {
57         result = token - '0';
58         match(token);
59     } else {
60         error();
61     }
62     return result;
63 }
64
65 void parse(void) {
66     getToken();
67     command();
68 }
69
70 int main() {
71     while (true) {
72         cout << "> ";
73         parse();
74     }
75     return 0;
76 }

```

```

45 // <expr> -> <term> (+ <term>, - <term>)
46 int expr(void) {
47     int result;
48     result = term();
49     while (token == '+' || token == '-') {
50         if (token == '+') {
51             match('+');
52             result += term();
53         } else if (token == '-') {
54             match('-');
55             result -= term();
56         }
57     }
58     return result;
59 }
60
61 // <term> -> <factor> (* <factor> | / <factor>) */
62 int term(void) {
63     int result = factor();
64     while (token == '*' || token == '/') {
65         if (token == '*') {
66             match('*');
67             result *= factor();
68         } else if (token == '/') {
69             match('/');
70             result /= factor();
71         }
72     }
73     return result;
74 }
75
76 // <factor> -> [-](<number> | (<expr>))
77 int factor(void) {
78     int result = 1;
79     if (token == '-') {
80         match('-');
81         result = -1;
82     }
83     if (token == '(') {
84         match('(');
85         result *= expr();
86         if (token == ')') {
87             match(')');
88         } else {
89             error();
90             exit(1);
91         }
92     } else {
93         result *= number();
94     }
95     return result;
96 }

```

```

fitt ~/Projects/algorithm_s
d/"p12-2
>> 4531 + (32 * 5) + (86 + 5)
4782
>> (69 * 8) + 911
1463
>> (74 + 88)
162
>> ((32*4)
syntax error!
d/"p12-2
>> 1000 + (( 1000 + 1000))
3000
>> 100 * ((100 + 100))
20000
>> 3 +
syntax error!

```

코드 설명

9~14: 아래에서 서로 호출하기 때문에 미리 선언

16~19: 문법 문제 발생 시 호출하는 함수

21~27: getchar() 함수를 이용해 입력 문자를 받아옴. 공백 문자 받으면 재귀호출

29~34: 파라미터와 토큰이 같으면 다음 토큰을 셋팅하는 함수 getToken() 호출. 다를 시 문법 에러

36~43: 명령어 expr()을 호출.expr() 호출 후 마지막 남은 input이 '\n'인 경우 정상. 아니면 에러

45~116: expr, term, factor, number, digit parsing function 구현

118~129: 메인함수. While문을 통해 에러 발생 전까지 계속 실행

Python, Java는 입출력 부분을 제외하고는 같은 로직입니다.

Python

```
3 token = ''
4 line = ''
5 cIdx = 0

13 def getchar():
14     global cIdx
15     c = line[cIdx]
16     cIdx += 1
17     return c
18
118 def main():
119     global line, token, cIdx
120     while(True):
121         token = ''
122         cIdx = 0
123         print(">> ", end='')
124         line = input() + '\n'
125         parse()

>> 4531 + (32 * 5) + (86 + 5)
4782
>> (69 * 8) + 911
1463
>> (74 + 88)
162
>> 1000 + ((1000+1000))
3000
>> 100 * ((100 + 100))
20000
>> 3 +
syntax error!
```

C++ 과 다르게 String으로 입력을 받고 index값을 이용하여 토큰을 받아옵니다.

Java

```
5 public static char token;
6 public static int cIdx;
7 public static String line;

9 public static void main(String[] args) {
10     Scanner scanner = new Scanner(System.in);
11     while(true) {
12         token = ' ';
13         cIdx = 0;
14         System.out.print(">> ");
15         line = scanner.nextLine() + '\n';
16         parse();
17     }
18 }

20 static void error() {
21     System.out.println("syntax error!");
22     exit( status: 1);
23 }

25 static char getchar(){
26     char c = line.charAt(cIdx++);
27     return c;
28 }

/Library/Java/JavaVirtualMachin
>> 4531 + (32 * 5) + (86 + 5)
4782
>> (69 * 8) + 911
1463
>> 1000 + ((1000+1000))
3000
>> 100 * ((100 + 100))
20000
>> 3 +
syntax error!
```

C++ 과 다르게 String으로 입력을 받고 index값을 이용하여 토큰을 받아옵니다.