

계산기 + 큰 수 프로젝트

CALCULATOR + BIGINT

BY 나종우, 양원우, 김수혁

개발 결과 요약

<https://github.com/njw1204/calculator-in-c>
(짧은주소 - <http://zipi.ga/q>)

- 계산기
 - 복잡한 수식을 입력해도 한 번에 계산 (사칙연산 + 괄호 지원)
 - ex) $1+(2*3)-4*5$ 입력 → -13 출력
 - 스택과 후위 표기식을 이용
- 큰 수 지원 (BigInt 구조체 구현)
 - $-10^{5000} \sim 10^{5000}$ (googol의 50제곱)
 - 기본 아이디어는 문자열 형태로 저장하는 것
- exit, clear 명령어 지원
 - exit : 프로그램 종료, clear : 화면 초기화

ULTIMATE CALCULATOR

You don't need to calculate one by one!

We can calculate the whole expression even which includes parentheses.

Support big integers between -10^{5000} and 10^{5000} .

The remainder of division is discarded.

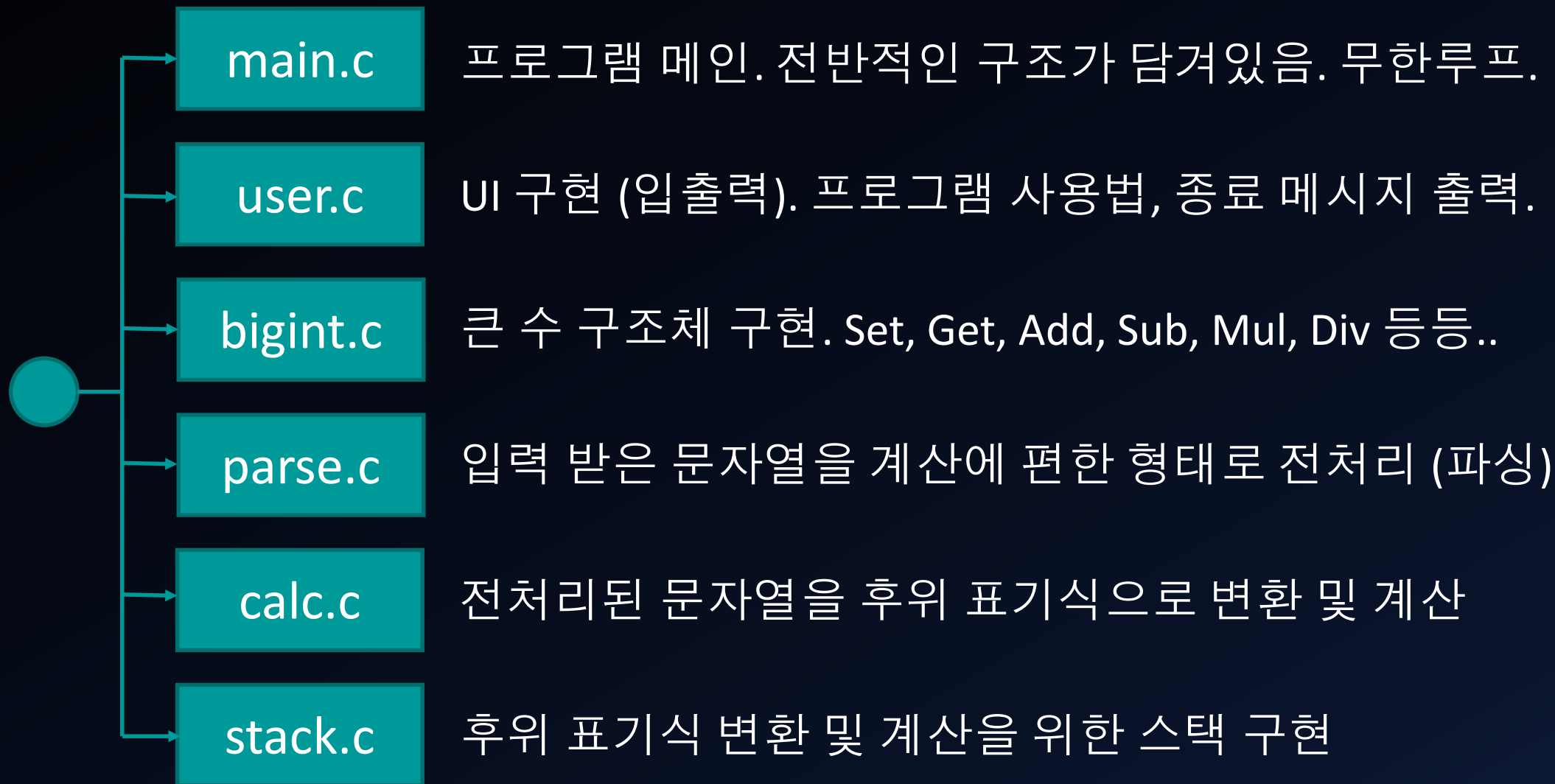
Input 'exit' to end, 'clear' to reset the screen.

```
> 1-3*(5-1)/2  
= -5
```

```
> -3240+672*(-12-(-5-((-31)*2+2)/(6-2*2)/1+0-(77))*99)/1+2)*((-7)*6)-1/1  
= 131464151
```

```
> 980-53535*((-10+(12345*(-6*5))-0)-5*4/10*(-6)*777)*5/50*(-9876543212345678987654321)  
= -19089444212179768411535648764601566
```

소스 구조



main.c

```
char exprRaw[50000];
CalcData expr[50000];
BigInt result;

while (1) {
    InputExp(exprRaw);
    ExpParse(exprRaw, expr, sizeof(expr));
    CalcExp(expr, &result);
    PrintResult(&result);
}
```

1) InputExp

→ char형 배열 exprRaw에 문자열 형태 수식을 입력 받음. (scanf)

2) ExpParse

→ exprRaw 배열에 들어있는 문자열을 계산하기 좋은 적절한 형태로 변환 후 CalcData형 배열 expr에 담음.

3) CalcExp

→ expr 배열에 담긴 수식을 계산한 후 결과값을 BigInt형 변수 result에 담음.

4) PrintResult

→ BigInt형 변수 result의 값을 출력.

BigInt 구현

```
typedef struct {  
    char sign;  
    int len;  
    char num[BIGINT_SIZE];  
} BigInt;
```

1) sign : 부호 (양수? 음수?)

→ 양수 : 1, 음수 : -1

2) len : 수의 길이

→ '12'의 길이는 2, '123'의 길이는 3

3) num : 수를 저장하는 문자 배열

→ 10진수의 형태로 저장

→ 계산의 용이함을 위해 거꾸로 저장

ex) 수가 '123654' 이면 "456321" 로 저장.

4) BIGINT_SIZE : BigInt의 표현 범위를 나타내는 상수

→ #define을 이용한 선언

BigInt 사칙연산 – 초등학교 방식대로

1. 덧셈, 뺄셈

- 초등학교에서 하던 방식, 흔히 사용하는 그 방식대로 구현
- 일의 자리부터 올라가며 차례대로 계산
- 올림 또는 내림 (carry) 이 발생하면 윗자리수로 carry를 넘김.

$$\begin{array}{r} 1 \\ 129 \\ + 28 \\ \hline 157 \end{array}$$

2. 곱셈, 나눗셈

- 역시 초등학교 방식, 흔히 쓰는 그 방식.

123

x 12

246

123

1476

→ 이런 방식으로 $99 * 12345$ 를 하면,

→ $(99*5) + (99*4)*10 + (99*3)*10*10 + \dots$

→ $x_0 \sim x_{10}$ 까지만 구현하면 모든 곱셈이 가능.

→ $x_0 \sim x_9$ 는 직접 더해서 구함.

→ x_{10} 은 맨 뒤에 0만 붙이면 끝.

나눗셈도 필산할때 쓰는
그 방법대로 구현하면 됩니다.
자세한 설명은 생략합니다.



Thank you