2023학년도 1학기 오픈소스소프트웨어프로젝트: Debugger과제

조교: 정재혁

학번/이름: 2018111750 / 이은학

[code 제출] 제공된 python 코드는 스택(Stack) 자료구조를 사용하여 +, *, -, / 연산자가 사용된 후위(postfix) 표기의 수식을 계산해주는 프로그램이지만, 에러가 발생한다.

아래 제공된 올바른 결과를 참고하여 Debugging을 통해 잘못된 부분을 수정하여라.

```
 ' 'c:\Users\kimmi\Desktop\hw_6\main_정답.py'
20 15 * 20 4 / + 8 * 5 -
2435.0
PS C:\Users\kimmi\Desktop\hw_6>
```

입력 데이터: 20 15 * 20 4 / + 8 * 5 -

출력 데이터: 2435.0

[1번째]

디버거 결과

```
13  for c in str:
14      x = 0
15      if c == '+':
16          x = acc.pop() + acc.pop()
17      elif c == '*':
18          x = acc.pop() * acc.pop()
```

```
예외가 발생했습니다. TypeError ×
can't multiply sequence by non-int of type 'str'
  File "C:\Users\leh\Desktop\OSSP\Debugger과제\main.py", line 18, in <module>
    x = acc.pop() * acc.pop()
TypeError: can't multiply sequence by non-int of type 'str'
```

해결 방법

str 타입끼리는 곱셈 연산을 할 수 없다는 TypeError 발생. 처음 문자열을 받을 때에 str형태로 받으므로 피연산자(자연수)들도 모두 str형태임을 인식. int형으로 바뀌어야하며, 다른 연산자들의 경우에도 마찬가지이므로 한번에 수정.

원인 코드

```
x = acc.pop()) + acc.pop()
x = acc.pop()) * acc.pop()
x = acc.pop()) - acc.pop()
x = acc.pop()) / acc.pop()
```

두 숫자를 pop하여 연산하는 수식 각 연산자별로 1줄씩 총 4줄

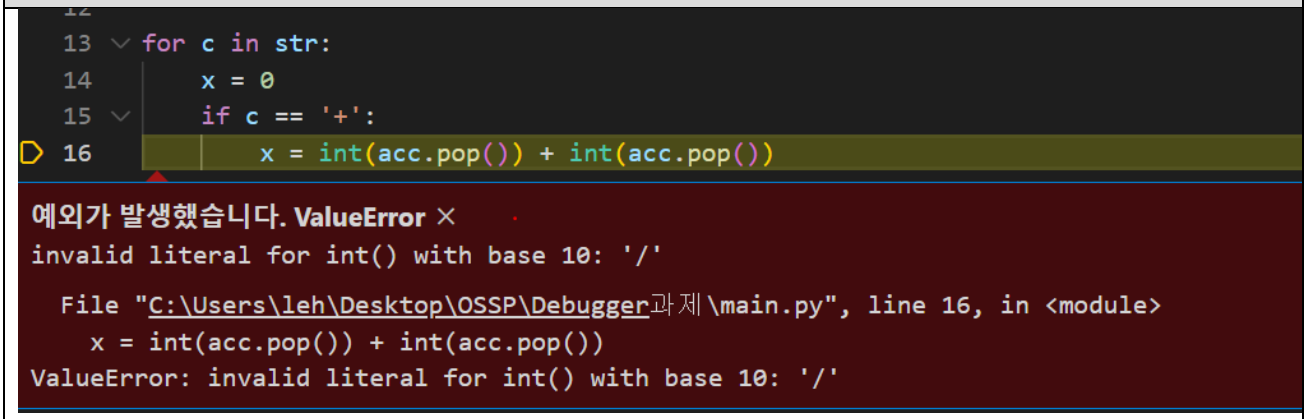
수정 코드

```
x = int(acc.pop()) + int(acc.pop())
x = int(acc.pop()) * int(acc.pop())
x = int(acc.pop()) - int(acc.pop())
x = int(acc.pop()) / int(acc.pop())
```

int()함수를 씌워 각각 피연산자를 정수형으로 변환

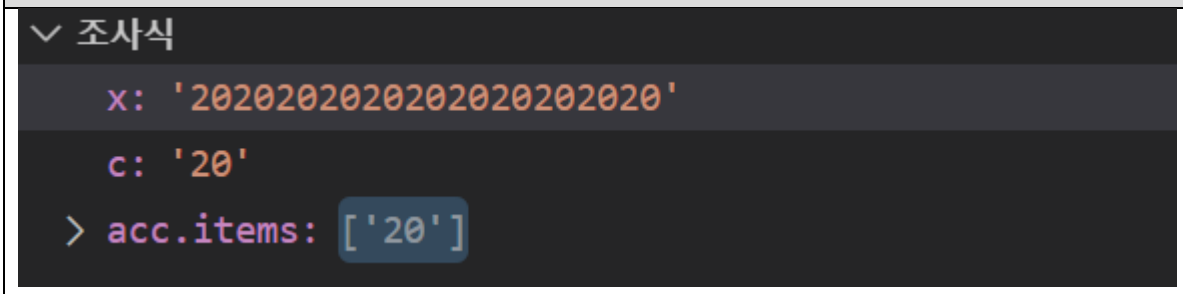
[2번째]

디버거 결과



'/' 기호를 int로 변환할 수 없다는 내용. acc.pop() 의 결과가 '/' , 즉, stack에 피연산자는 없고 연산자만 담겨있었다는 뜻. 잘못된 push, pop 연산이 일어나고 있음을 인식

해결 방법



for문을 돌면서 x와 c와 stack에 쌓이는 item을 acc.items로 조사.
첫 번째 for문(c='20'일 경우)을 돌았을 때의 조사식을 보면
x에 20이 아니라 20을 11번 붙인 문자열이 들어가있음 => 계산식 오류

원인 코드

```
    elif c >= '0' and c <= '9':
        x = 10 * c + c
```

c에 피연산자(자연수)가 들어갈 경우 str형태임을 고려한 조건식으로 보이지만, 계산식이 왜 저런지는 이해가 안됨 ...

수정 코드

```
    else :
        x = c
```

그러나 어차피 연산자 외에는 모두 피연산자이므로 else로 조건식 수정.

[3번째]

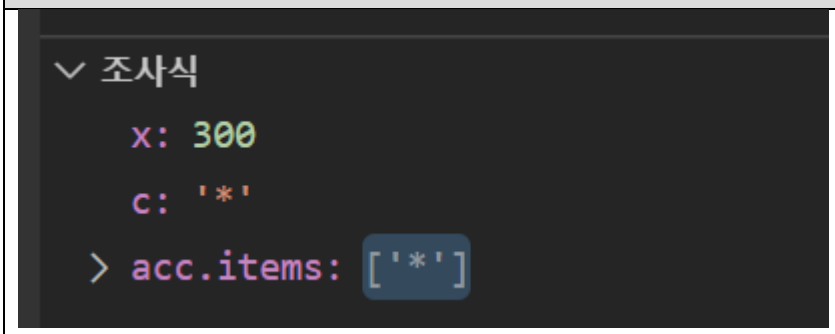
디버거 결과

```
13  for c in str:
14      x = 0
15      if c == '+':
16          x = int(acc.pop()) + int(acc.pop())
```

```
예외가 발생했습니다. ValueError ×
invalid literal for int() with base 10: '/'
  File "C:\Users\leh\Desktop\OSSP\Debugger과제\main.py", line 16, in <module>
    x = int(acc.pop()) + int(acc.pop())
ValueError: invalid literal for int() with base 10: '/'
```

같은 에러 발생. 문제가 아직 남아있음

해결 방법



for문 세 번째 바퀴를 돌고 난 경우, stack에 20, 10이 사라지고 300이 생기는 대신 *만 생김 x에 300이 들어간걸로 보아 pop 후 * 연산까지는 문제가 없고, push 연산에서 문제가 있는 것으로 보임

원인 코드

```
acc.push(c)
```

연산 결과를 x에 저장해놓고 왜 c를 push?

수정 코드

```
acc.push(x)
```

x를 push하는 것으로 수정

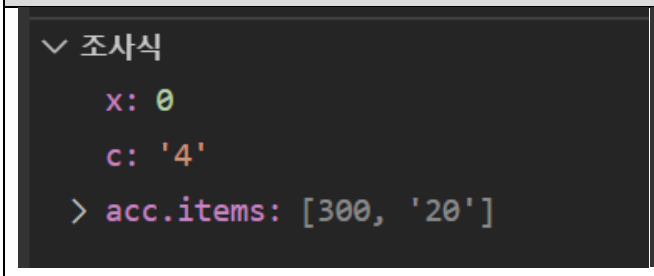
[4번째]

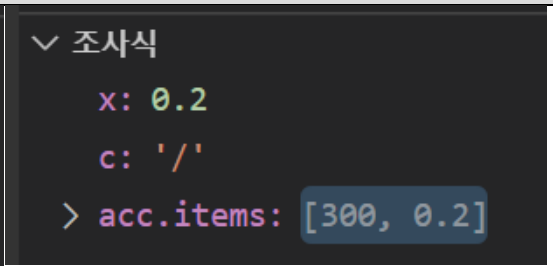
디버거 결과

```
20 15 * 20 4 / + 8 * 5 -
-2395
```

에러는 없지만 결과값이 틀림. 계산 식에서 오류가 있는지 찾기 위해 다시 한줄씩 debugging

해결 방법





c='/' 일 때, 20/4 연산결과가 5가 아닌 0.2가 되었음.

원인 코드

```
    elif c == '/':
        x = int(acc.pop()) / int(acc.pop())
```

4가 먼저 pop되고 20이 pop되므로 연산 순서가 바뀌었음을 인식

수정 코드

```
    elif c == '/':
        temp1 = int(acc.pop())
        temp2 = int(acc.pop())
        x = temp2 / temp1
```

임시 변수 temp1, temp2 생성하여 연산 순서를 바꿔주는 코드로 수정

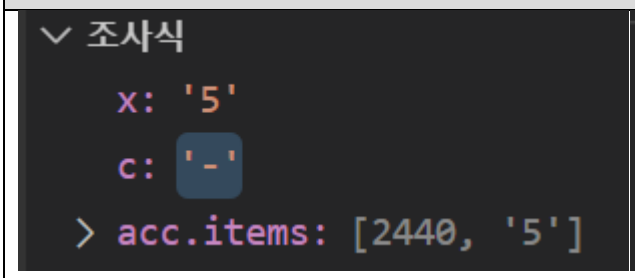
[5번째]

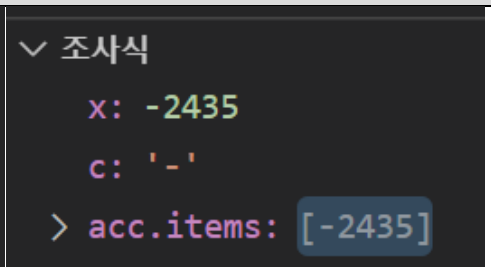
디버거 결과

```
20 15 * 20 4 / + 8 * 5 -
-2435
```

계산결과 오류 : 부호 반대

해결 방법





c='-' 일 때, x가 2440 – 5가 아닌 5 – 2440 이 된 것을 보고, 나눗셈의 경우와 마찬가지로 연산순서가 바뀌었음을 알 수 있음

원인 코드

```
    elif c == '-':
        x = int(acc.pop()) - int(acc.pop())
```

수정 코드

```
    elif c == '-':
        temp1 = int(acc.pop())
        temp2 = int(acc.pop())
        x = temp2 - temp1
```

나눗셈 연산과 같은 방법으로 수정

[6번째]

디버거 결과

```
20 15 * 20 4 / + 8 * 5 -
2435
```

소수점이 표시되지 않은 모습. 과제의 목표는 2435.0 을 출력하는 것이다

해결 방법

소수점이 표시되는 이유는, 나눗셈 연산 20 / 5 에서 소수점이 생겼을 텐데, 내 코드에서는 push하기 전에 int로 변환해주기 때문에 없어진 것으로 파악했다.
연산 후에 int로 변환해주는 과정이 필요없으려면, 애초에 피연산자를 push할 때에 int형으로 바꾸어서 저장하면 되겠다고 생각하여 첫 번째 과정에서 수정했던 코드를 다시 되돌리고, push할 때에 int()함수를 사용해준다.

원인 코드

```
x = int(acc.pop()) + int(acc.pop())
x = int(acc.pop()) * int(acc.pop())
x = int(acc.pop()) - int(acc.pop())
x = int(acc.pop()) / int(acc.pop())
```

연산식마다 붙였던 int()함수

수정 코드

```
x = acc.pop()) + acc.pop()
x = acc.pop()) * acc.pop()
x = acc.pop()) - acc.pop()
x = acc.pop()) / acc.pop()
```

다시 되돌려주고

```
    else :
        x = int(c)
```

c = 피연산자일 경우에 int로 변환한 후에 push하도록 수정

[최종 코드]

최종 코드

```python
class stack:
    def __init__(self):  # 스택 객체 생성
        self.items = []
    def push(self, item):  # 스택 요소 추가 push(.append())
        self.items.append(item)
    def pop(self):   # 스택 요소 삭제 pop()
        return self.items.pop()

acc = stack()
str = input().split()
x = 0

for c in str:
    x = 0
    if c == '+':
        x = acc.pop() + acc.pop()
    elif c == '*':
        x = acc.pop() * acc.pop()
    elif c == '-':
        temp1 = acc.pop()
        temp2 = acc.pop()
        x = temp2 - temp1
    elif c == '/':
        temp1 = acc.pop()
        temp2 = acc.pop()
        x = temp2 / temp1
    else :
        x = int(c)
    acc.push(x)

x = acc.pop()
print(x)
```