Postfix Evaluation Example
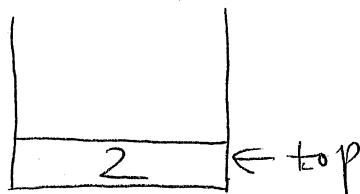
postfix: 2 3 * 4 +
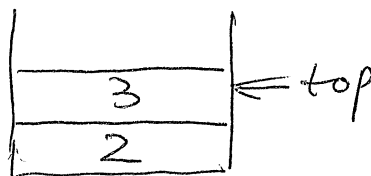
while Loop first iteration
    current item is '2', is   operand;
    push ('2');

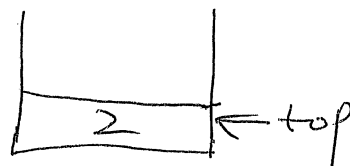| |
|---|
| 2 | ← top

while Loop second iteration
    current is '3', is operand
    push('3')

3	← top
2

while Loop third iteration
    current is '*' is operator
    pop stack,   right = 3

2	← top

    pop stack   left = 2

empty
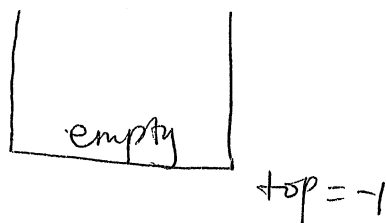
top = -1

    evaluate: push(left * right)
                push(6)

6	← top

while Loop Fourth iteration:
    current item is '4', is operand.
        push('4')

4	← top
6
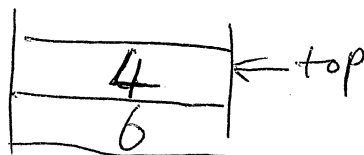
while Loop fifth iteration

Current item is '+' ; operator

right = pop() = 4

$$6 \leftarrow top$$

left = pop() = 6

empty
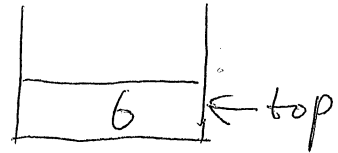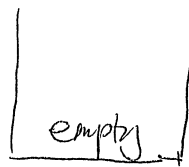
top = -1

evaluation

push (left '+' right)
push (10)

$$10 \leftarrow top$$
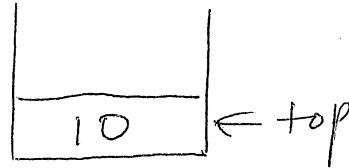
Then while Loop drops out !

only one number on the stack; is 10,
10 is the final result for the original postfix expression.

Note here :
① We learned that items on stack usually have the same data type.

② Here, we assume the operands and operators are stored in a string, and operands have only one digit. What if an operands have arbitrary number of digits?

③ How to solve this problem when you implement this algorithm?