

Ref1. 실험(Case 1) 처음 고객 요구 수식

IDX=0

번호	PostExp	OpExp
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1		
0	3	

Check = false PrevIDX=0
Check = True

IDX=1

번호	PostExp	OpExp
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1		
0	3	+

Check = True PrevIDX=0
Check = false PrevIDX=1

1. PrevData가 널이 아닐 때
FALSE
2. ISP(+) ICP(3) 진행안함
3. PrevIDX = 1
4. OpExp->Push(+)
5. Check = false

IDX=2

번호	PostExp	OpExp
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1	4	
0	3	+

Check = False PrevIDX=1
Check = True

IDX=3

번호	PostExp	OpExp
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2	5	
1	4	
0	3	+

Check = True PrevIDX=1

IDX=4

번호	PostExp	OpExp
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3	+	
2	5	
1	4	
0	3	-

Check = True PrevIDX=1
Check = False PrevIDX=4

1. PrevData가 널이 아닐 때
TRUE
ISP(-) ICP(+) 진행함
0 3
ISP(-) <ICP(+)
TRUE
2. OpExp->POP()
3. Postexp->Push()
4. PrevIDX = 4로 바꾸고
5. OpExp->Push(-)
6. Check = false

IDX=5

번호	PostExp	OpExp
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3	+	
2	5	
1	4	-
0	3	-

Check = False PrevIDX=4

IDX=6

번호	PostExp	OpExp
13		
12		
11		
10		
9		
8		
7		
6		
5		

IDX=7

번호	PostExp	OpExp
13		
12		
11		
10		
9		
8		
7		
6		
5	8	

IDX=8

번호	PostExp	OpExp
13		
12		
11		
10		
9		
8		
7		
6	~	
5	8	

4	7	
3	+	
2	5	
1	4	-
0	3	-

Check = False PrevIDX=4
Check = True

4	7	
3	+	
2	5	
1	4	-
0	3	-

Check = True PrevIDX=4

4	7	
3	+	
2	5	
1	4	*
0	3	-

Check = True PrevIDX=4 개선전
Check = False PrevIDX=8

```
1. PrevData가 널일 때 True
{
    2. OpExp->POP()
    3. Postexp->Push()
    4. PrevIDX = 8로 바꾸고
    5. OpExp->Push(*)
    6. Check = false
    7. continue
}
```

1. PrevData가 널이 아닐 때

```
ISP(*)      ICP(-)    진행함
2            3
TRUE
```

IF (ISP(PrevData) < ICP(NextData)) 참일 때
{

```
2. if(PrevIDX에 속한 데이터 범위에서 -가 2이상일때
{
    while(OpExp->GetTop() < 0)
    {
        if ( PrevData = GetData(GetTop() - 1)이 음수일 때
        {
            2. OpExp->POP()
            3. Postexp->Push()
        }else 탈출;
    }
}
else
{
    while(OpExp->GetTop() < -1)
    OpExp->POP()
    Postexp->Push()
}
}
}else{
    ISP 비교 거짓
    2. OpExp->POP()
    3. Postexp->Push()
    4. PrevIDX = 8로 바꾸고
    5. OpExp->Push(*)
    6. Check = false
}
```

IDX=9

번호	PostExp	OpExp
13		
12		
11		
10		
9		
8		
7	5	
6	~	
5	8	
4	7	
3	+	
2	5	
1	4	*
0	3	-

Check = False PrevIDX=8
Check = True

IDX=10

번호	PostExp	OpExp
13		
12		
11		
10		
9	-	
8	*	
7	5	
6	~	
5	8	
4	7	
3	+	
2	5	
1	4	
0	3	-

Check = True PrevIDX=8 개선전
Check = False PrevIDX=10

```
1. PrevData가 널일 때 True
{
    2. OpExp->POP()
    3. Postexp->Push()
    4. PrevIDX = 8로 바꾸고
```

```

5. OpExp->Push(*)
6. Check = false
7. continue

}
1. PrevData가 널이 아닐 때

ISP(*)      ICP(-)   진행함
  2          3

TRUE

IF ( ISP(PrevData) < ICP(NextData)) 참일 때
{
    2. if(PrevIDX에 속한 데이터 범위에서 -가 2이상일때
    {
        while(OpExp->GetTop() < 0)
        {
            if ( PrevData = GetData(GetTop() - 1)이 음수일 때
            {
                2. OpExp->POP()
                3. Postexp->Push()
            }else 탈출;
        }
    }
    else
    {
        while(OpExp->GetTop() < -1)
        {
            OpExp->POP()
            Postexp->Push()
        }
        4. PrevIDX = 10로 바꾸고
        5. OpExp->Push(*)
        6. Check = false
    }
}
}
}else{
    ISP 비교 거짓
    2. OpExp->POP()
    3. Postexp->Push()
    4. PrevIDX =
    5. OpExp->Push(*)
    6. Check = false
    continue
}
}

```

IDX=11

번호	PostExp	OpExp
13		
12		
11		
10	2	
9	-	
8	*	
7	5	
6	~	
5	8	
4	7	
3	+	
2	5	
1	4	
0	3	-

Check = False PrevIDX=10
Check = True

IDX=12

번호	PostExp	OpExp
13		
12		
11		
10	2	
9	-	
8	*	
7	5	
6	~	
5	8	
4	7	
3	+	
2	5	
1	4	^
0	3	-

Check = True PrevIDX=10 개선전
Check = False PrevIDX=12

```

1. PrevData가 널일 때 True
{
    2. OpExp->POP()
    3. Postexp->Push()
    4. PrevIDX = 8로 바꾸고
    5. OpExp->Push(*)
    6. Check = false
    7. continue
}

```

1. PrevData가 널이 아닐 때

ISP(^) ICP(-) 진행함
2 3
TRUE

IF (ISP(PrevData) < ICP(NextData)) 참일 때

```
{
    2. if(PrevIDX에 속한 데이터 범위에서 -가 2이상일때
    {
        while(OpExp->GetTop() < 0)
        {
            if ( PrevData = GetData(GetTop() - 1)이 음수일 때
            {
                2. OpExp->POP()
                3. Postexp->Push()
            }else 탈출;
        }
    }
    else
    {
        if(NextData != '^')
            while(OpExp->GetTop() < -1)

    }
    4. PrevIDX = 12로 바꾸고
    5. OpExp->Push(^)
    6. Check = false
    continue
}
}
}else{
    ISP 비교 거짓
    2. OpExp->POP()
    3. Postexp->Push()
    4. PrevIDX =
    5. OpExp->Push(*)
    6. Check = false
    continue
}
```

IDX=13

번호	PostExp	OpExp
13		
12		
11	3	
10	2	
9	-	
8	*	
7	5	
6	~	
5	8	
4	7	
3	+	
2	5	
1	4	^
0	3	-

Check = False PrevIDX=12
Check = True

IDX=14

번호	PostExp	OpExp
13		
12	^	
11	3	
10	2	
9	-	
8	*	
7	5	
6	~	
5	8	
4	7	
3	+	
2	5	
1	4	*
0	3	-

Check = True PrevIDX=12 개선2차
Check = False PrevIDX=14

1. PrevData가 널일 때 True

```
{
    2. OpExp->POP()
    3. Postexp->Push()
    4. PrevIDX = 8로 바꾸고
    5. OpExp->Push(*)
    6. Check = false
    7. continue
}
```

1. PrevData가 널이 아닐 때

ISP(*) ICP(^) 진행함
1 1

FALSE

```
IF ( ISP(PrevData) < ICP(NextData)) 참일 때
{
    2. if(PrevIDX에 속한 데이터 범위에서 -가 2이상일때
    {
        while(OpExp->GetTop() < 0)
        {
            if ( PrevData = GetData(GetTop() - 1)이 음수일 때
            {
                2. OpExp->POP()
                3. Postexp->Push()
            }else 탈출;
        }
    }
    else
    {
        if(NextData != '^')
            while(OpExp->GetTop() < -1)

        }
        4. PrevIDX = 12로 바꾸고
        5. OpExp->Push(^)
        6. Check = false
        continue
    }
}
}else{
    ISP 비교 거짓
    2. OpExp->POP()
    3. Postexp->Push()
    4. PrevIDX = 14
    5. OpExp->Push(*)
    6. Check = false
    continue
}
```

IDX=15

번호	PostExp	OpExp
16		
15		
14		
13		
12	^	
11	3	
10	2	
9	-	
8	*	
7	5	
6	~	
5	8	
4	7	
3	+	
2	5	(
1	4	*
0	3	-

Check = False PrevIDX=14
PrevIDX=15

```
if(NextData == '(')
    PrevIDX = 15
```

IDX=17

번호	PostExp	OpExp
16		
15		

IDX=16

번호	PostExp	OpExp
16		
15		
14		
13	5	
12	^	
11	3	
10	2	
9	-	
8	*	
7	5	
6	~	
5	8	
4	7	
3	+	
2	5	(
1	4	*
0	3	-

Check = False PrevIDX=15 개선3차

IDX=18

번호	PostExp	OpExp
16		
15		

14		
13	5	
12	^	
11	3	
10	2	
9	-	
8	*	
7	5	
6	~	
5	8	
4	7	
3	+	-
2	5	(
1	4	*
0	3	-

Check = True PrevIDX=15 개선4차
Check = False PrevIDX=17

14	2	
13	5	
12	^	
11	3	
10	2	
9	-	
8	*	
7	5	
6	~	
5	8	
4	7	
3	+	-
2	5	(
1	4	*
0	3	-

Check = False PrevIDX=17
Check = True PrevIDX=17

1. PrevData가 널일 때 True

```
{
    2. OpExp->POP()
    3. Postexp->Push()
    4. PrevIDX = 8로 바꾸고
    5. OpExp->Push(*)
    6. Check = false
    7. continue
}
```

1. PrevData가 널이 아닐 때

ISP(-) ICP(*) 진행함
0 2
TRUE

참고

Check == false 아래가

NextData == ')'판단이 있어야함.
그래서 잔량 데이터 다 처리해야함

IF (ISP(NextData) < ICP(PrevData)) 참일 때

```
{
    2. if(PrevIDX에 속한 데이터 범위에서 -가 20이상일때
    {
        while(OpExp->GetTop() < 0)
        {
            if ( PrevData = GetData(GetTop() - 1)이 음수일 때
            {
                2. OpExp->POP()
                3. Postexp->Push()
            }else 탈출;
        }
    }
    else
    {
        if(NextData != '^')
            while(OpExp->GetTop() < -1)
                PrevData = OpExp->GetData(OpExp->GetTop());

        if(ISP(PrevData) == 괄호번호)
            break;
        else
            PrevData = OpExp->Pop();
            Postexp->Push(PrevData);
    }

    4. PrevIDX = 12로 바꾸고
    5. OpExp->Push(NextData)
    6. Check = false
    continue
}

}

}else{
    ISP 비교 거짓
    2. OpExp->POP()
    3. Postexp->Push()
    4. PrevIDX = 14
    5. OpExp->Push(*)
}
```

```
6. Check = false  
continue
```

```
}
```