

객체지향프로그래밍

1-2

1. 2개의 수를 입력 받아 두 수의 크기를 구분하는 프로그램을 구현하시오. 프로그램은 두 수 A,B를 입력받아 A가 B보다 클 경우 '**A > B**'를, 같을 경우 '**A = B**'를, B가 A보다 클 경우 '**A < B**'를 출력한다.

```
Enter Two Number :5 4
```

<입력 예시>

```
Enter Two Number :5 4  
5 > 4
```

<출력 예시>

2. 숫자 N ($0 < N < 20$)을 입력 받아 1부터 20까지 N번 째 수에서 줄을 바꾸며 출력하는 프로그램을 구현하시오.

```
Enter any Number :4  
1      2      3      4  
5      6      7      8  
9      10     11     12  
13     14     15     16  
17     18     19     20
```

<4를 입력 받았을 때>

```
Enter any Number :3  
1      2      3  
4      5      6  
7      8      9  
10     11     12  
13     14     15  
16     17     18  
19     20
```

<3을 입력 받았을 때>

3. 3x3의 보드에서 Command를 입력 받아 Node가 움직이는 프로그램을 구현하시오. 보드는 모두 '0'으로 채워져 있으며 Node는 'x'로 표기한다. 프로그램이 시작 시 Node는 좌상단 끝(0.0)에 위치하고 있으며 Command의 입력에 의해 움직인다. 프로그램은 시작 시 보드 및 Node를 출력하고 Command를 입력받을 때마다 재출력하는 방식으로 동작한다.

Command는 아래와 같으며 만약 Node가 명령어에 의해 보드의 범위를 넘어가려 하면 입력을 무시한다.

a : Move Left	w : Move Up
d : Move Right	s : Move Down
q : Finish	

<Command 목록>

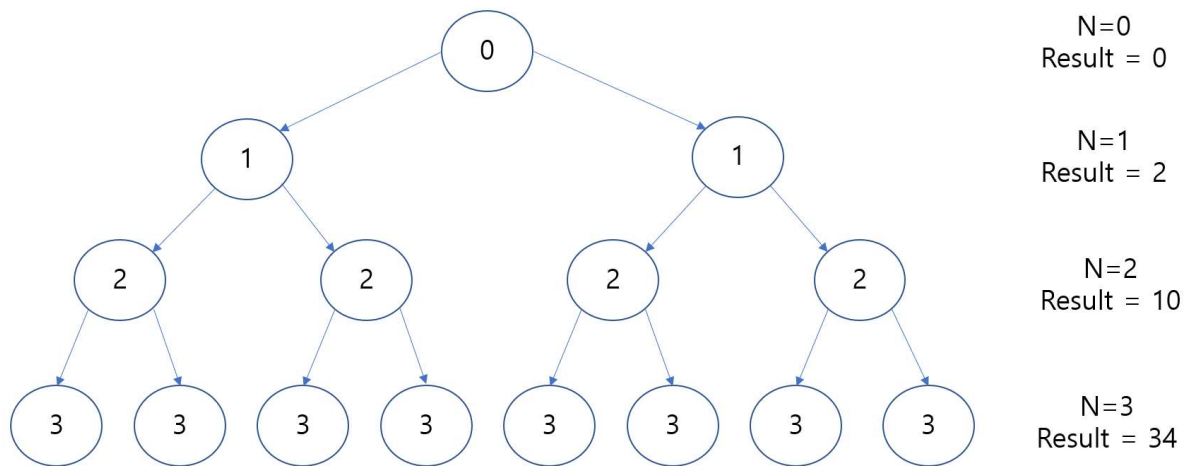
```

x      0      0
0      0      0
0      0      0
Enter Move Command(a : Left, d: Right, w: Up, s: Down, q: finish) :d
0      x      0
0      0      0
0      0      0
Enter Move Command(a : Left, d: Right, w: Up, s: Down, q: finish) :s
0      0      0
0      x      0
0      0      0
Enter Move Command(a : Left, d: Right, w: Up, s: Down, q: finish) :a
0      0      0
x      0      0
0      0      0
Enter Move Command(a : Left, d: Right, w: Up, s: Down, q: finish) :w
x      0      0
0      0      0
0      0      0
Enter Move Command(a : Left, d: Right, w: Up, s: Down, q: finish) :q

```

<프로그램 예시>

4. 양의 정수 N을 입력 받아 $\sum_{i=0}^N (i * 2^i)$ 을 계산하는 프로그램을 구현하시오. 프로그램은 시작 숫자 N을 입력 받으며 숫자에 따른 계산결과를 출력한다. 이 때 반복문(for, while, do while) 등은 사용하지 않고 함수의 호출을 이용해 구현한다.



<프로그램의 함수 호출 구조>

```

Enter Any Positive Number : 4
98

```

<프로그램의 입력 및 출력>