

Homework 6

SNU 4910.210 Spring 2006

due: 6/12 24:00

Exercise 1 “파도”

게임(예, 고스톱)을 진행하다가 중간에 문제가 생기면 “이번 판은 무효”로 진행을 멈추게 됩니다. 예외상황(exception)으로 다루어지는 경우입니다.

다음의 두 갈래 나무구조 tree를 생각하자:

```
type tree = Node of int * tree * tree
          | Leaf of int
          | Empty
```

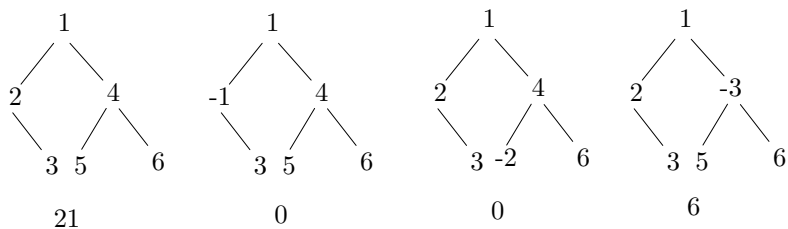
함수 sum을 생각하자:

```
sum : tree → int
```

함수 sum은 입력 나무의 노드에 기록된 정수들을 모두 합하는 데, 다음과 같이 합해야 한다:

- sum(t) 계산 중에 왼쪽 하부트리(left sub-tree)의 뿌리에서 만나는 정수값이 음수이면 sum(t)의 최종 결과는 0.
- sum(t) 계산 중에 오른쪽 하부트리(right sub-tree)의 뿌리에서 만나는 정수값이 음수이면 그 하부트리의 sum값은 0으로 한다.

예를 들어 다음의 나무구조들에서 sum의 값은 각각 21, 0, 0, 6이다.



위의 조건을 만족하도록 `sum`의 두 버전을 정의하라. `sum-straight`는 예외 상황을 사용하지 않고 정의하고, `sum-exception`은 예외 상황을 이용해서 작성하라. □

Exercise 2 “이길로”

nML에서는 예외 상황을 일으킬 때 임의의 값을 예외 상황에 싣고 예외 상황 처리기로 전달할 수 있다.

위의 `sum-exception`을 다시 정의해서, 다음의 조건이 만족하도록 하라:

- `sum-exception(t)` 계산 중에 왼쪽 하부트리(left sub-tree)의 뿌리에서 만나는 정수값이 음수이면 최종 결과는 0.
- `sum-exception(t)` 계산 중에 오른쪽 하부트리(right sub-tree)의 뿌리에서 만나는 정수값이 음수이면 그 하부트리의 `sum-exception`값은 그 음수값으로 한다.

위의 세 가지 나무구조들에서 위의 `sum-exception`을 돌리면 각각 21, 0, 0, 3을 계산한다. □