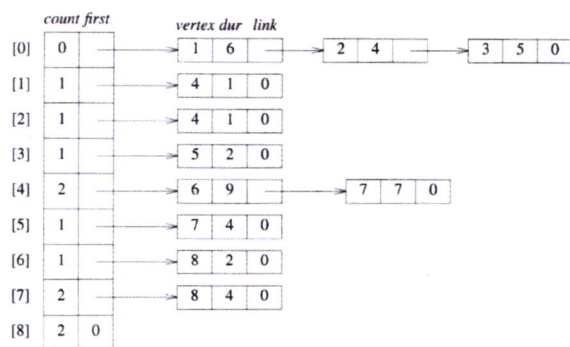
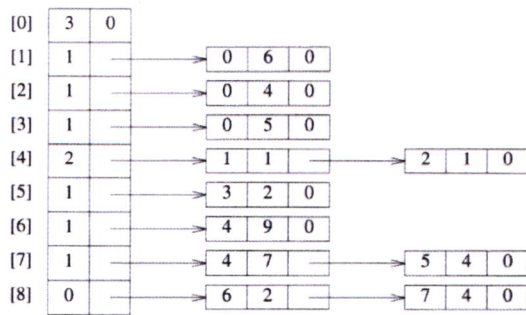


1. 다음 AOE 네트워크에 대하여 `setup4()` 함수를 수정하여 입력처리한다. => 소스코드만 제출한다.

-  $ee[j] = \max_{i \in P(j)} \{ee[i] + \text{duration of } \langle i, j \rangle\}$  => ee[j]를 출력할 때 P(j)를 출력한다 /이 부분이 없으면 0점 처리

-  $le[j] = \min_{i \in S(j)} \{le[i] - \text{duration of } \langle j, i \rangle\}$  => le[j]를 출력할 때 S(j)를 출력한다/이 부분  
이 없으면 0점 처리

[illegible]



le	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	Stack
initial	18	18	18	18	18	18	18	18	18	[8]
output 8	18	18	18	18	18	18	16	14	18	[7,6]
output 7	18	18	18	18	7	10	16	14	18	[5,6]
output 5	18	18	18	8	7	10	16	14	18	[3,6]
output 3	3	18	18	8	7	10	16	14	18	[6]
output 6	3	18	18	8	7	10	16	14	18	[4]
output 4	3	6	6	8	7	10	16	14	18	[2,1]
output 2	2	6	6	8	7	10	16	14	18	[1]
output 1	0	6	6	8	7	10	16	14	18	[0]

## 2. tree 표현 문제이다. => 소스 코드만 제출한다.

- 다음과 같은 infix notation 을 연산자 우선 순위에 입각하여 binary tree 를 만들기 위한 insert 함수를 구현한다.
- e = "A + B \* C - D + E \* F / G - H + K - L"를 prefix 또는 postfix 로 변환한 string 변수 ex 를 char 하나씩 가져와 binary tree 를 만든다.
- 변수는 char로 선언한다.
- 연산자는 + - \* / 만 사용하며 연산자 우선순위에 따라, 동등한 우선 순위이면 오른쪽부터 왼쪽 순서로 priority 가 적용된다.(이 부분 처리 안하면 0 점)
- 입력된 연산 expression 에 대하여 inorder, preorder, postorder, level order traversal 결과를 출력한다.
- 소스코드를 compile 하여 실행하면 입력된 e 와 변환된 ex 를 보이고 tree 를 traversal 한 결과를 보여야 한다.