### (훈련반2) Level29

## 그래프 / 트리의 탐색의 두번째 방법인 BFS (Breadth Deapth Search, 너비우선탐색)을 시작 해 봅시다.

## DFS와 달리, **while과 큐**를 이용하여 구현합니다.

## 그래프를 탐색하는 두 가지 방법 DFS와 BFS 을 모두 익혀봅시다.

## **Level29 대문자만 DFS**

**문제 1번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1034)]

문자열로 구성된 이진트리를 입력 받아주세요.

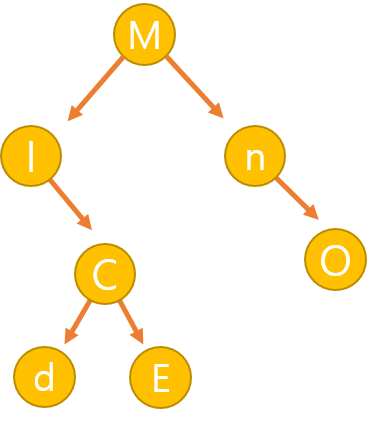
(1차원 배열에 저장하는 형태로 입력이 주어집니다.)

(#은 노드가 없음을 의미합니다.)

Root노드부터 DFS를 돌리면서

**대문자 노드를 탐색할 때마다 출력 해 주세요.**

만약 #MIn#C#O##dE을 입력받았다면



출력결과 : MICEO

## 입력 예제

#MIn#C#O##dE

## 출력 결과

MICEO





## Level29 문자열 노드 DFS

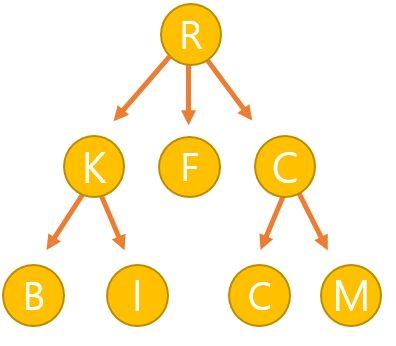
**문제 2번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1034)]

8개의 노드로 구성 된 문자열과

대응되는 인접행렬을 입력받아주세요.

아래 이미지는 입력 예시에 해당하는 트리입니다.

0번 노드부터 DFS로 노드들을 탐색하면서 출력 해 주세요.



## 입력 예제

RKFCBICM

0 1 1 1 0 0 0 0

0 0 0 0 1 1 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 1 1

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

## 출력 결과

RKBIFCCM





## Level29 링크드리스트와 DFS

**문제 3번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1034)]

**6글자를 입력받고** **링크드리스트로 다음과 같이 연결시켜주세요.**

ex)

head = new Node( );

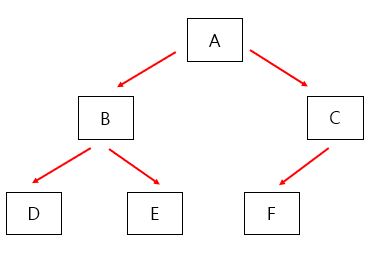
head->left = new Node( );

head->right = new Node( );

...

링크드리스트 노드에 각각 입력받은 문자를 넣어주면 됩니다.

만약, ABCDEF 를 입력 받았다면 아래와 같습니다.



**이제 DFS를 돌린결과를 출력 해주세요.**

## 입력 예제

ABCDEF

## 출력 결과

ABDECF



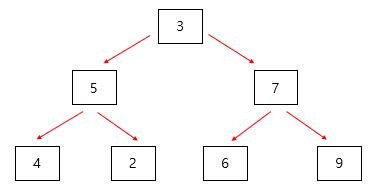
## **Level29 도착 경로의 합**

**문제 4번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1034)]

**노드에 들어갈 숫자** **7개를** **입력** **받으세요.**

만약 **3 5 7 4 2 6 9** 를 입력 받았다면 다음과 같습니다.

(Root는 1번 index에 저장해야 함을 잊지마세요)



이제 **마지막 노드(Level2)에 있는 숫자들의 합을**

**DFS를 돌려 구해주세요.**

## 입력 예제

ㄴ

## 출력 결과

21



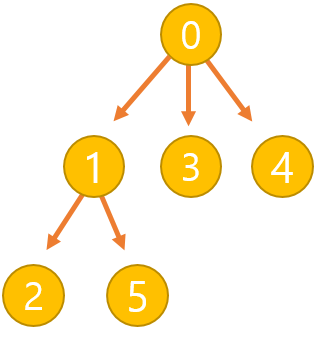
## Level29 홀수만 BFS

**문제 5번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1034)]

0 ~ 5번까지 6개 노드로 구성된 인접행렬을 입력받아주세요.

0번 노드부터 BFS를 돌려, **홀수 노드를 찾을 때 마다 출력** 해 주세요.

만약 위와 같이 트리를 입력받았다면,



**출력결과 : 1 3 5**

## 입력 예제

0 1 0 1 1 0

0 0 1 0 0 1

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

## 출력 결과

1 3 5





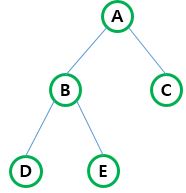
## Level29 TREE DFS

**문제 6번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1034)]

노드 안에 들어갈 문자 5개를 입력 받으세요.

만약 ABCDE 입력 받으면 아래와 같이 저장됩니다.

(트리모양은 고정입니다)



위 트리를 **인접행렬**로 저장하고, DFS를 돌려 탐색 순서대로 출력하세요.

(힌트 : 입력받은 문자열은 배열에 따로 저장해두어야 합니다)

## 입력 예제

ABCDE

## 출력 결과

ABDEC

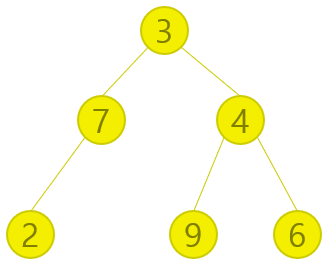


## **Level29 이진트리 BFS**

**문제 7번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1034)]

총 8개의 숫자를 1차원 배열에 입력받습니다. **(1~9 사이 숫자, 0은 없는 노드)**

BFS를 돌려 **탐색 순서대로 값을 출력** 해주세요.



만약 **0 3 7 4 2 0 9 6** 을 입력 받았다면 위와 같은 트리가 됩니다.

ex) **0 3 7 4 2 0 9 6** =>  **출력결과 : 3 7 4 2 9 6**

## 입력 예제

0 1 2 3 0 0 4 5

## 출력 결과

1 2 3 4 5



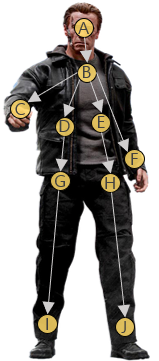


## Level29 터미네이터 신경망

**문제 8번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1034)]

터미네이터의 신경체계가 다음과 같이 트리형태로 되어있다고 합니다.

인접행렬 형태로 초기화를 해 주세요.



이 신경쳬계를 트리로 하드코딩 하고, 나노 탐사로봇이 탐색을 하려고 합니다.

BFS 알고리즘을 쓸때 탐사순서대로 출력해주세요.

예를들어 E를 입력받았다면,

E에서 출발하는 BFS를 돌리면 됩니다. (출력결과 : E H J)

예를들어 B를 입력받았다면,

B에서 출발하는 BFS를 돌리면 됩니다. (출력결과 : B C D E F G H I J)

## 입력 예제

A

## 출력 결과

A B C D E F G H I J



## Level29 번식왕 두더지

**문제 9번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1034)]

조상 두더지는 1년에 세마리의 자식을 낳습니다.

1년 후 각각의 자식들은 각자 세마리의 자식을 낳습니다.

**큐를 이용해서 n년 후에는 총 몇마리의 두더지가 있는지 출력** 해주세요.



ex)

0년 => 1마리

1년 => 3마리 + 1마리 = 4마리

2년 => 9마리 + 3마리 + 1마리 = 13마리

[힌트]

Queue에 1을 넣어두고 시작합니다.

Head index의 값에 \* 3 곱한 값을 큐에 추가하고, Head++ (큐에 3이 적혀집니다)

Head index의 값에 \* 3 곱한 값을 큐에 추가하고, Head++ (큐에 9가 적혀집니다)

이 과정을 n번 반복한 후

큐에 적혀있는 숫자들을 모두 더하면 됩니다.

## 입력 예제

1

## 출력 결과

4



