재귀함수

재귀함수는 자기 자신을 호출하는 함수를 의미합니다.

```
def DFS(n):
    if n == 0:
        return
    else:
        print(n, end = ' ')
        DFS(n-1)

DFS(5)
```

위에 DFS 함수는 자신의 함수 안에서 자기 자신을 호출하고 있습니다. 이런 함수를 재귀함수라고 합니다.
위에 프로그램을 실행시키면 화면에 5, 4, 3, 2, 1순으로 출력됩니다.

```
def DFS(n):
    if n == 0:
        return
    else:
        DFS(n-1)
        print(n, end = ' ')

DFS(5)
```

그런데 print(n, end = ' ') 코드라인을 DFS(n-1) 아래로 이동하고 실행시키면 화면에
1, 2, 3, 4, 5순으로 출력되는 것을 알 수 있습니다.
이렇게 print()의 위치를 함수호출 아래로 옮겼을 때 출력순서가 반대로 나오는 것은 재귀함수가 스택을 사용해서 작동하기 때문입니다.

n! 구하기

매개변수 n에 자연수가 입력되면 n!(n팩토리얼) 값을 구하여 반환하는 프로그램을 작성하세요. 예를 들어 n = 5라면 5! = 5 * 4 * 3 * 2* 1 = 120을 반환하면 됩니다.

입출력 예:

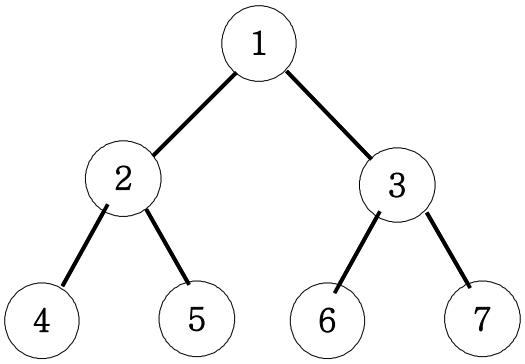
n	answer
5	120
6	720
7	5040
8	40320

제한사항:

- 1 <= n <= 10

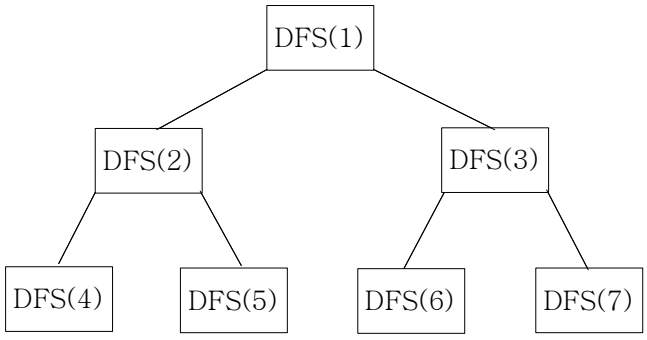
이진트리 깊이우선탐색

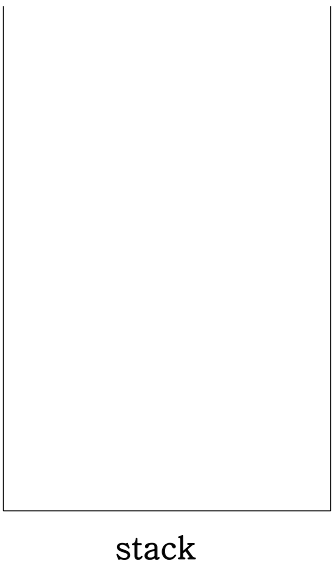
아래 그림과 같은 이진트리를 깊이우선탐색을 해보세요.



깊이우선탐색 출력 : 1 2 4 5 3 6 7

```
0       def DFS(v):
1           if v > 7:
2               return
3           else:
4               print(v, end=' ')
5               DFS(v*2)
6               DFS(v*2+1)
7
8       DFS(1)
```





피보나치 수열

피보나치 수열의 n번째 항은 n-1번째 항과 n-2번째 항을 합으로 구합니다.
피보나치 수열의 첫 번째 항과 두 번째 항은 1입니다.
피보나치 수열을 구해보면
1, 1, 2, 3, 5, 8, 13, 21, 34, ...
입니다.
매개변수 n에 구해야 할 피보나치 수열의 항 번호가 주어지면 피보나치 수열의 n번째 항을 구해 반환하는 프로그램을 작성하세요.

입출력 예:

n	answer
5	5
6	8
8	21
10	55

제한사항:
- 1 <= n <= 12

검정색 영역구하기

5 * 5 이차원 배열에 모니터 화면을 표현합니다. 모니터의 화변은 최초 검정색과 흰색으로만 표현되어 있습니다. 검정색은 1, 흰색은 0으로 표현됩니다.

	0	1	2	3	4
0	0	1	1	0	0
1	0	1	1	0	0
2	0	1	0	0	0
3	0	0	0	1	0
4	0	0	1	1	0

상하좌우로 1(검정색)이 연결되어 있으면 한 영역으로 간주합니다.
화면의 격자 정보가 위와 같다면 검정색으로 칠해진 영역은 2곳입니다.
매개변수 board에 모니터 화면의 격자정보가 주어지면 검정색으로 칠해진 영역은 총 몇 개가 있는지 구하여 반환하는 프로그램을 작성하세요.

입출력 예:

board	answer
[[0, 1, 1, 0, 0], [0, 1, 1, 0, 0], [0, 1, 0, 0, 0], [0, 0, 0, 1, 0], [0, 0, 1, 1, 0]]	2
[[1, 1, 1, 0, 1], [1, 1, 1, 0, 1], [0, 0, 1, 0, 0], [1, 1, 0, 1, 0], [1, 0, 1, 0, 0]]	5
[[0, 0, 1, 0, 0], [0, 1, 1, 0, 0], [0, 1, 0, 0, 0], [1, 0, 0, 1, 0], [0, 0, 1, 1, 0]]	3
[[0, 0, 0, 0, 1], [0, 0, 1, 0, 0], [0, 1, 0, 1, 0], [0, 0, 0, 1, 0], [0, 0, 1, 0, 0]]	5

제한사항:
- 검정색 영역은 1개 이상 반드시 존재합니다.

픽셀수구하기

5 * 5 이차원 배열에 모니터 화면을 표현합니다. 모니터의 화변은 최초 검정색과 흰색으로만 표현되어 있습니다. 검정색은 1, 흰색은 0으로 표현됩니다.

	0	1	2	3	4
0	0	1	1	0	0
1	0	1	1	0	0
2	0	1	0	0	0
3	0	0	0	1	0
4	0	0	1	1	0

상하좌우로 1(검정색)이 연결되어 있으면 한 영역으로 간주합니다.
화면의 격자 정보가 위와 같다면 검정색으로 칠해진 영역은 2이고, 첫 번째 영역의 픽셀수(격자수)는 5개이고, 두 번째 영역의 픽셀수는 3개입니다.
매개변수 board에 모니터 화면의 격자정보가 주어지면 검정색으로 칠해진 각 영역의 픽셀수를 순서대로 배열에 담아 반환하세요. 영역의 순서는 각 영역의 행번호, 열번호가 가장 작은 픽셀을 기준으로 행번호가 작은 것부터이며 행번호가 같을 경우 열 번호가 작은 영역 순으로 배열에 담습니다.

입출력 예:

board	answer
[[0, 1, 1, 0, 0], [0, 1, 1, 0, 0], [0, 1, 0, 0, 0], [0, 0, 0, 1, 0], [0, 0, 1, 1, 0]]	[5, 3]
[[1, 1, 1, 0, 1], [1, 1, 1, 0, 1], [0, 0, 1, 0, 0], [1, 1, 0, 1, 0], [1, 0, 1, 0, 0]]	[7, 2, 3, 1, 1]
[[0, 0, 1, 0, 0], [0, 1, 1, 0, 0], [0, 1, 0, 0, 0], [1, 0, 0, 1, 0], [0, 0, 1, 1, 0]]	[4, 1, 3]
[[0, 0, 0, 0, 1], [0, 0, 1, 0, 0], [0, 1, 0, 1, 0], [0, 0, 0, 1, 0], [0, 0, 1, 0, 0]]	[1, 1, 1, 2, 1]

제한사항:
• 검정색 영역은 1개 이상 반드시 존재합니다.