

재귀함수를 이용한 이진수 출력

10진수 N 이 입력되면 2진수로 변환하여 출력하는 프로그램을 작성하세요. 단 재귀함수를 이용해서 출력해야 합니다.

▣ 입력설명

매개변수 n 에 10진수 $N(1 \leq N \leq 1,000)$ 이 주어집니다.

▣ 출력설명

n 의 이진수를 반환하세요.

▣ 매개변수 형식

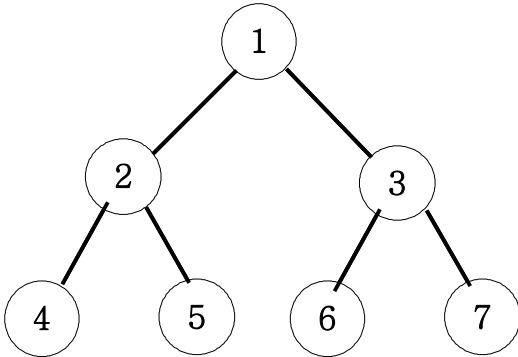
11

▣ 반환값 형식

1011

이진트리 순회(깊이우선탐색 : DFS)

아래 그림과 같은 이진트리를 전위순회와 후위순회를 연습해보세요.



전위순회 출력 : 1 2 4 5 3 6 7

중위순회 출력 : 4 2 5 1 6 3 7

후위순회 출력 : 4 5 2 6 7 3 1

아래 코드는 위에 있는 이진트리를 전위순회한 것입니다. 여러분이 아래 코드를 분석해보고, 중위순회, 후위순회를 출력해보세요. 그리고 스택에 스택프레임을 만들면서 분석도 해보세요.

```
function solution(n){
    let answer="";
    function DFS(v){
        if(v>7) return;
        else{
            answer+=(v+' ');
            DFS(v*2);
            DFS(v*2+1);
        }
    }
    DFS(n);
    return answer;
}
console.log(solution(1));
```

부분집합 구하기(DFS)

자연수 N 이 주어지면 1부터 N 까지의 원소를 갖는 집합의 부분집합을 모두 출력하는 프로그램을 작성하세요.

■ 입력설명

매개변수 n 에 자연수 $N(1 \leq N \leq 10)$ 이 주어집니다.

■ 출력설명

부분집합의 각 경우를 배열로 반환합니다.

단 공집합은 반환하지 않습니다.

■ 매개변수 형식 1

3

■ 반환값 형식

[[1, 2, 3], [1, 2], [1, 3], [1], [2, 3], [2], [3]]

합이 같은 부분집합

N개의 원소로 구성된 자연수 집합이 주어지면, 이 집합을 두 개의 부분집합으로 나누었을 때 두 부분집합의 원소의 합이 서로 같은 경우가 존재하면 "YES"를 출력하고, 그렇지 않으면 "NO"를 출력하는 프로그램을 작성하세요.

둘로 나뉘는 두 부분집합은 서로소 집합(Disjoint Set)이며, 두 부분집합을 합하면 입력으로 주어진 원래의 집합이 되어야 합니다.

예를 들어 {1, 3, 5, 6, 7, 10}이 입력되면 {1, 3, 5, 7} = {6, 10} 으로 두 부분집합의 합이 16으로 같은 경우가 존재하는 것을 알 수 있다.

■ 입력설명

매개변수 nums에 $N(1 \leq N \leq 10)$ 개의 원소로 구성된 집합이 주어집니다. 각 원소는 중복되지 않으며, 집합의 총 합은 짝수로 주어집니다.

■ 출력설명

"YES" 또는 "NO"를 반환합니다.

■ 매개변수 형식 1

[1, 3, 5, 6, 7, 10]

■ 반환값 형식 1

YES

■ 매개변수 형식 2

[5, 2, 6, 9, 10, 12]

■ 반환값 형식 2

YES

■ 매개변수 형식 3

[3, 9, 11, 13]

■ 반환값 형식 3

NO

바둑이 승차(DFS)

철수는 그의 바둑이들을 데리고 시장에 가려고 한다. 그런데 그의 트럭은 C킬로그램 넘게 태울수가 없다. 철수는 C를 넘지 않으면서 그의 바둑이들을 가장 무겁게 태우고 싶다. N마리의 바둑이와 각 바둑이의 무게 W가 주어지면, 철수가 트럭에 태울 수 있는 가장 무거운 무게를 구하는 프로그램을 작성하세요.

■ 입력설명

매개변수 nums에 $N(1 \leq N \leq 30)$ 마리 바둑이의 무게 정보가 주어집니다.

매개변수 c에 자연수 $C(1 \leq C \leq 100,000,000)$ 가 주어집니다.

■ 출력설명

가장 무거운 무게를 반환하세요.

■ 매개변수 형식 1

[81, 58, 42, 33, 61], 259

■ 반환값 형식 1

242

■ 매개변수 형식 2

[34, 56, 55, 67, 33, 76, 63, 43], 379

■ 반환값 형식 2

372

최대점수 구하기(DFS)

이번 정보올림피아드대회에서 좋은 성적을 내기 위하여 현수는 선생님이 주신 N 개의 문제를 풀려고 합니다. 각 문제는 그것을 풀었을 때 얻는 점수와 푸는데 걸리는 시간이 주어지게 됩니다. 제한시간 M 안에 N 개의 문제 중 최대점수를 얻을 수 있도록 해야 합니다. (해당문제는 해당시간이 걸리면 푸는 걸로 간주한다, 한 유형당 한개만 풀 수 있습니다.)

■ 입력설명

매개변수 $nums$ 에 $N(1 \leq N \leq 20)$ 개의 문제 정보가 주어집니다.

매개변수 m 에 제한 시간 $M(10 \leq M \leq 300)$ 이 주어집니다.

■ 출력설명

제한 시간안에 얻을 수 있는 최대 점수를 반환합니다.

■ 매개변수 형식 1

[[10, 5], [25, 12], [15, 8], [6, 3], [7, 4]], 20

■ 반환값 형식 1

41

■ 매개변수 형식 2

[[15, 6], [30, 11], [23, 8], [14, 4], [10, 3], [20, 7]], 25

■ 반환값 형식 2

74

중복순열 구하기

1부터 N 까지 번호가 적힌 구슬이 있습니다. 이 중 중복을 허락하여 M 번을 뽑아 일렬로 나열하는 방법을 모두 출력합니다.

■ 입력설명

매개변수 n 에 자연수 $N(3 \leq N \leq 10)$ 이 주어지고, 매개변수 m 에 $M(2 \leq M \leq N)$ 이 주어집니다.

■ 출력설명

중복순열의 경우를 배열 형태로 반환합니다. 각 경우의 순서는 오름차순입니다.

■ 매개변수 형식 1

3 2

■ 반환값 형식 1

[[1, 1], [1, 2], [1, 3], [2, 1], [2, 2], [2, 3], [3, 1], [3, 2], [3, 3]]

순열 구하기

10이하의 N 개의 자연수가 주어지면 이 중 M 개를 뽑아 일렬로 나열하는 방법을 모두 출력합니다.

■ 입력설명

매개변수 $nums$ 에 $N(3 \leq N \leq 10)$ 개의 자연수가 오름차순으로 주어집니다.

매개변수 m 에 $M(2 \leq M \leq N)$ 이 주어집니다.

■ 출력설명

중복순열의 결과를 배열형태로 반환합니다.

■ 매개변수 형식 1

[3, 6, 9], 2

■ 반환값 형식 1

[[3, 6], [3, 9], [6, 3], [6, 9], [9, 3], [9, 6]]

조합의 경우수(메모이제이션)

$nCr = \frac{n!}{(n-r)!r!}$ 로 계산합니다. 하지만 여러분은 이 공식을 쓰지않고 아래 공식을 사용하여 재귀를 이용해 조합수를 구해주는 프로그램을 작성하세요.

$${}_nC_r = {}_{n-1}C_{r-1} + {}_{n-1}C_r$$

▣ 입력설명

매개변수 n에 N(3<=N<=33)과 매개변수 r에 R(0<=R<=n)이 입력됩니다.

▣ 출력설명

조합수를 반환합니다.

▣ 매개변수 형식 1

5 3

▣ 반환값 형식 1

10

▣ 매개변수 형식 2

33 19

▣ 반환값 형식 2

818809200

수열 추측하기

가장 윗줄에 1부터 N까지의 숫자가 한 개씩 적혀 있다. 그리고 둘째 줄부터 차례대로 파스칼의 삼각형처럼 위의 두개를 더한 값이 저장되게 된다. 예를 들어 N이 4 이고 가장 윗 줄에 3 1 2 4 가 있다고 했을 때, 다음과 같은 삼각형이 그려진다.

```

3 1 2 4
4 3 6
7 9
16

```

N과 가장 밑에 있는 숫자가 주어져 있을 때 가장 윗줄에 있는 숫자를 구하는 프로그램을 작성하시오. 단, 답이 여러가지가 나오는 경우에는 사전순으로 가장 앞에 오는 것을 출력하여야 한다.

입력설명

매개변수 n에 $N(1 \leq N \leq 10)$ 과 매개변수 f에 F가 주어집니다.

N은 가장 윗줄에 있는 숫자의 개수를 의미하며 F는 가장 밑에 줄에 있는 수로 1,000,000 이하입니다.

출력설명

삼각형에서 가장 위에 들어갈 N개의 숫자를 배열형태로 반환합니다. 답이 존재하지 않는 경우는 입력으로 주어지지 않는다.

매개변수 형식 1

4 16

반환값 형식 1

[3, 1, 2, 4]

매개변수 형식 2

5 50

반환값 형식 2

[1, 2, 4, 3, 5]

조합 구하기

1부터 N 까지 번호가 적힌 구슬이 있습니다. 이 중 M 개를 뽑는 방법의 수를 출력하는 프로그램을 작성하세요.

■ 입력설명

매개변수 n 에 자연수 $N(3 \leq N \leq 10)$ 과 매개변수 m 에 $M(2 \leq M \leq N)$ 이 주어집니다.

■ 출력설명

조합의 결과를 배열형태로 반환하세요.

■ 매개변수 형식 1

4 2

■ 반환값 형식 1

[[1, 2], [1, 3], [1, 4], [2, 3], [2, 4], [3, 4]]

수들의 조합

N개의 정수가 주어지면 그 숫자들 중 M개를 뽑는 조합의 합이 임의의 정수 K의 배수인 개수는 몇 개가 있는지 출력하는 프로그램을 작성하세요.

예를 들면 5개의 숫자 2 4 5 8 12가 주어지고, 3개를 뽑은 조합의 합이 6의 배수인 조합을 찾으면 4+8+12 2+4+12로 2가지가 있습니다.

■ 입력설명

매개변수 nums에 $N(3 \leq N \leq 20)$ 개의 정수가 주어집니다.

매개변수 m에 자연수 $M(2 \leq M \leq N)$ 이 주어지고, 매개변수 k에 K값이 주어집니다.

■ 출력설명

총 가지수를 반환합니다.

■ 매개변수 형식 1

[2, 4, 5, 8, 12], 3, 6

■ 반환값 형식 1

2

■ 매개변수 형식 2

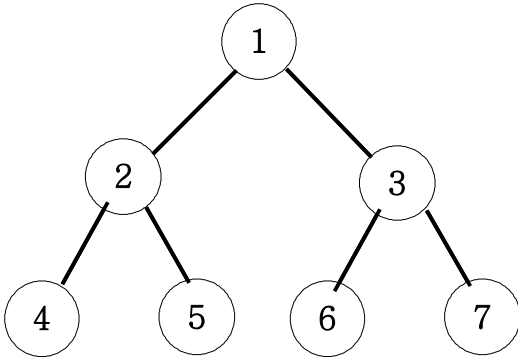
[3, 5, 7, 8, 9, 12, 14], 4, 8

■ 반환값 형식 2

5

이진트리 레벨탐색(넓이우선탐색 : BFS)

아래 그림과 같은 이진트리를 큐(Queue) 자료구조를 이용해 레벨탐색을 해보세요.



레벨탐색 출력 : 1 2 3 4 5 6 7

아래 코드는 Queue 자료구조를 이용해 위에 트리를 레벨탐색한 코드입니다. 스스로 분석해 보세요.

```
function solution(){
  let answer="";
  function BFS(){
    let queue=[];
    queue.push(1);
    while(queue.length){
      let v=queue.shift();
      answer+=v+" ";
      for(let nv of [v*2, v*2+1]){
        if(nv>7) continue;
        queue.push(nv);
      }
    }
  }
  BFS();
  return answer;
}
console.log(solution());
```

송아지 찾기(BFS : 상태트리탐색)

현수는 송아지를 잃어버렸다. 다행히 송아지에는 위치추적기가 달려 있다. 현수의 위치와 송아지의 위치가 수직선상의 좌표 점으로 주어지면 현수는 현재 위치에서 송아지의 위치까지 다음과 같은 방법으로 이동한다. 송아지는 움직이지 않고 제자리에 있다.

현수는 스카이 콩콩을 타고 가는데 한 번의 점프로 앞으로 1, 뒤로 1, 앞으로 5를 이동할 수 있다. 최소 몇 번의 점프로 현수가 송아지의 위치까지 갈 수 있는지 구하는 프로그램을 작성하세요.

■ 입력설명

매개변수 s에 현수의 위치 S와 매개변수 e에 송아지의 위치 E가 주어집니다.
직선의 좌표 점은 1부터 10,000까지이다.

■ 출력설명

점프의 최소횟수를 반환합니다. 답은 1이상입니다.

■ 매개변수 형식 1

5 14

■ 반환값 형식 1

3

■ 매개변수 형식 2

8 3

■ 반환값 형식 2

5

미로의 최단거리 통로(BFS)

7*7 격자판 미로를 탈출하는 최단경로의 길이를 출력하는 프로그램을 작성하세요. 경로의 길이는 출발점에서 도착점까지 가는데 이동한 횟수를 의미한다. 출발점은 격자의 (1, 1) 좌표이고, 탈출 도착점은 (7, 7)좌표이다. 격자판의 1은 벽이고, 0은 도로이다.

격자판의 움직임은 상하좌우로만 움직인다. 미로가 다음과 같다면

출발	0	0	0	0	0	0
0	1	1	1	1	1	0
0	0	0	1	0	0	0
1	1	0	1	0	1	1
1	1	0	1	0	0	0
1	0	0	0	1	0	0
1	0	1	0	0	0	도착

위와 같은 경로가 최단 경로의 길이는 12이다.

■ 입력설명

매개변수 board에 7*7 격자의 정보가 주어집니다.

■ 출력설명

최단으로 움직인 칸의 수를 반환합니다. 도착할 수 없으면 -1를 출력한다.

■ 입력예제 1

```
[[0, 0, 0, 0, 0, 0, 0], [0, 1, 1, 1, 1, 1, 0], [0, 0, 0, 1, 0, 0, 0], [1, 1, 0, 1, 0, 1, 1],
[1, 1, 0, 1, 0, 0, 0], [1, 0, 0, 0, 1, 0, 0], [1, 0, 1, 0, 0, 0, 0]]
```

■ 출력예제 1

12