[임베디드 트랙] 3회차 과목평가 – 알고리즘

| Background

✓ 알고리즘 기법을 이해하고 문제를 해결

| Goal

✓ 알고리즘을 이용하여 문제를 해결, 그 결과를 표현할 수 있다.

| 환경 설정

1) 프로그램 언어 선택 : C, C++

2) 라이브러리 사용 가능

3) 제출 파일 이름 형식 예시

: 알고리즘1번_서울_15반_홍길동.cpp, 알고리즘2번_서울_15반_홍길동.cpp

4) 제출 파일

: 작성한 파일을 "알고리즘_서울_반 번호_이름.zip" 형태로 압축하여 제출한다.

: 예시 : 알고리즘_서울_15반_홍길동.zip

5) 테스트 케이스 : 모든 테스트 케이스는 공개되지 않으며, 부분적으로 제공된다.

6) 채점 : 테스트 케이스별로 부분 채점 된다.

성실과 신뢰로 테스트에 임할 것 (부정 행위시 강력 조치 및 근거가 남음)

※ 소스코드 유사도 판단 프로그램 기준 부정 행위로 판단될 시, 0점 처리 및 학사 기준에 의거 조치 실시 예정

[임베디드 트랙] 3회차 과목평가 – 알고리즘

| Problem 01 : 마법사의 몬스터 사냥(배점 : 60점)

시간 : C, C++ 1초

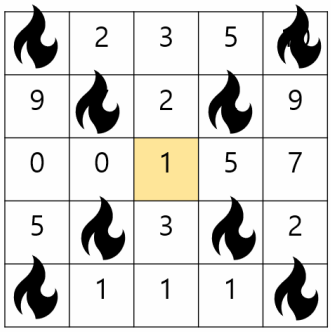
N x N 사각형의 전투장에는 각 칸마다 몇 마리의 몬스터가 있는지 적혀 있다.

광대한 영역에 마법을 시전할 수 있는 마법사 baldMort 는 전투장에서 최대한 많은 몬스터를 잡으려한다.

마법사 baldMort는 대각선↗, ↙, ↖, ↘ 방향으로 각각 K 칸 만큼 마법을 시전할 수 있다.

1	2	3	5	10
9	7	2	2	9
0	0	1	5	7
5	2	3	2	2
1	1	1	1	1

[그림1]

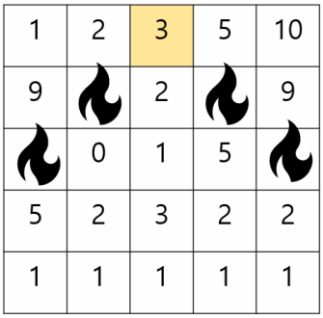


[그림2]

예를들어 [그림1]와 같은 5 x 5 인 전투장에서 노란색으로 표시된 2번 행 2번 열에서 K = 2 인 마법을 시전하게되면 각 방향마다 2칸씩 , [그림2] 와 같이 몬스터를 공격하게 되며 총 1 + 10 + 7 + 2 + 2 + 2 + 1 + 1 = 26 마리를 처치하게 된다.

[임베디드 트랙] 3회차 과목평가 – 알고리즘

[그림3]



[그림4]

반면에 [그림3]와 같이 0번 행 2번 열에서 K = 2인 마법을 시전하면 [그림4] 와 같이 몬스터를 공격하게 되며 총 7 + 2 + 0 + 7 = 16 마리 몬스터를 처치할 수 있다.

마법사 baldMort 씨가 처치할수 있는 몬스터의 최대 수를 출력하시오.

[입력]

첫째줄에 전투장의 가로세로크기인 N 이 입력된다. (1 <= N <= 100)

다음 줄부터는 N 줄에 걸쳐 각줄마다 N개의 정수가 공백으로 구분되어 입력된다.

(0<= 정수 <= 100000)

마지막 줄에는 마법의 시전범위 K 가 입력된다. (1 <= K <= 100)

[출력]

마법사가 잡을 수 있는 몬스터의 최대 수를 출력하시오.

[입력 예시]

5

1 2 3 5 10

9 7 2 2 9

0 0 1 5 7

5 2 3 2 2

1 1 1 1 1

2

[출력 예시]

26

[임베디드 트랙] 3회차 과목평가 – 알고리즘

| Problem 02 : N을 만드는 수식(배점 : 30점)

시간 : C, C++ 1초

1부터 N까지의 정수들을 활용하여 합을 N으로 만들 수 있는 경우의 수를 구하시오.

[제한 조건]

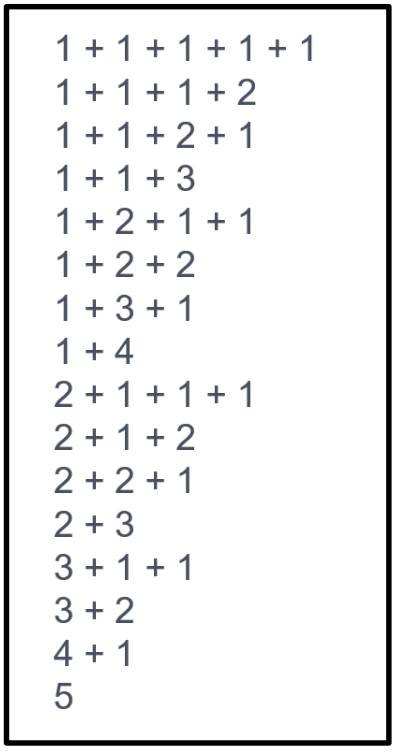
- 중복된 조합은 다시 카운팅하지 않습니다.
- 같은 정수는 최대 2번까지 사용할 수 있습니다.

[예시]

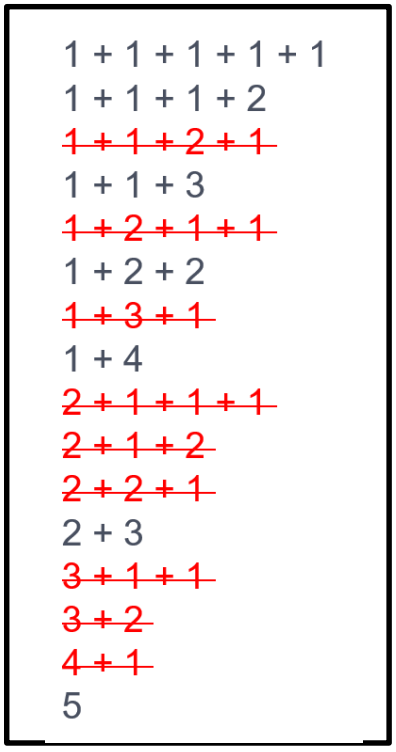
N=5 일때, 1부터 5까지의 정수들을 가지고 합을 N=5로 만들 수 있는 경우의 수는 **[그림 1]**과 같습니다.

여기서 첫번째 제한 조건을 적용했을때에는, **[그림 2]**와 같은 경우의 수들이 남습니다.

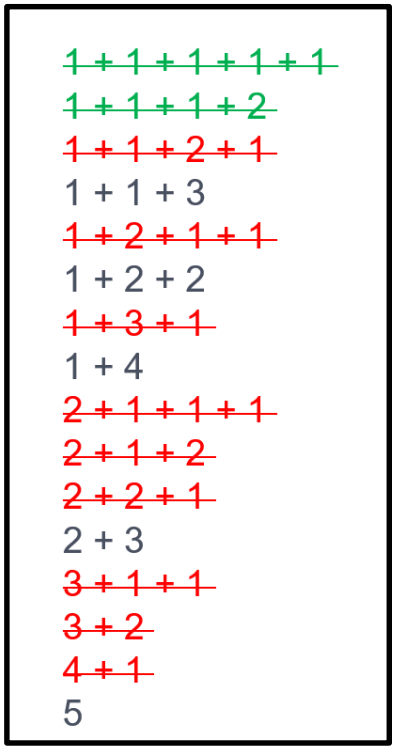
그리고 마지막 제한 조건을 적용하면 **[그림 3]**과 같은 경우의 수만 남습니다.



[그림 1]



[그림 2]



[그림 3]

최종적으로 유효한 경우의 수는 5개로, **5**를 출력합니다.

[임베디드 트랙] 3회차 과목평가 – 알고리즘

[입력]

첫번째 줄에 정수 N을 입력받습니다. (1 <= N <= 50)

[출력]

첫번째 줄에 1부터 N을 활용한 합으로 N을 만들 수 있는 경우에 수 중 제한 조건에 부합하는 유효한 조합의 수를 출력합니다.

[입력 예시1]

5

[출력 예시1]

5

[입력 예시2]

1

[출력 예시2]

1

[입력 예시3]

10

[출력 예시3]

22

[임베디드 트랙] 3회차 과목평가 – 알고리즘

| Problem 03 : 서술형 문제 (배점 : 10점)

DFS와 BFS는 그래프의 탐색 알고리즘입니다.
이 두 알고리즘의 차이를 상세히 서술해주세요.