

오늘은, 다음 문제들을 풀어보고 부족한 곳 스스로 찾아서 그쪽 부분 문제를 더 풀어보시거나, 선행하셔도 됩니다.

1.

설명	'q'가 입력될 때까지 입력한 문자를 계속 출력하는 프로그램을 작성해보자.
입력예시	x b k d l q g a c
출력예시	x b k d l q

2.

설명	정수(1~100) 1개를 입력받아 1부터 그 수까지 짝수의 합을 구해보자.
입력예시	5
출력예시	6

3.

설명	시작 값(a), 등비(r), 몇 번째인지를 나타내는 정수(n)가 입력될 때 n번째 수를 출력하는 프로그램을 만들어보자.
입력	시작 값(a), 등비의 값(r), 몇 번째 인지를 나타내는 정수(n)가 공백을 두고 입력된다.(모두 0 ~ 10)
출력	n번째 수를 출력한다.
입력예시	2 3 7
출력예시	1458

4.

설명	3 6 9 게임을 하던 용빈이는 3 6 9 게임에서 잦은 실수로 계속해서 벌칙을 받게 되었다. 3 6 9 게임의 왕이 되기 위한 마스터 프로그램을 작성해 보자.
입력	자연수 1개가 입력된다.
출력	1부터 그 수까지 순서대로 출력하는데, 게임 369에서 박수를 쳐야하는 숫자 대신 영문 대문자 'X'를 출력한다.
입력예시	13
출력예시	1 2 X 4 5 X 7 8 X 10 11 12 X (공백이 아니라 줄바꿈이여도 됨)

5.

설명	한 생선가게에는 머루를 포함해 고양이 여러마리가 들어와 물고기들을 훔쳐간다. 귀여움을 참다 못한 주인장이, 가장 고양이들이 많이 온 날 한 번에 고양이들을 잡아서 집에 데려가서 키우려고 한다. 고양이들이 언제 오는지 확인해보았더니, 매우 규칙적이게 움직이는 것이 아닌가! n 마리의 고양이들이 각각 m_1, m_2, \dots, m_n 의 주기를 가지고 올 때, 며칠이 지나 서 고양이를 잡아야 할까?
입력	m_1, m_2, \dots, m_n 가 공백을 기준으로 주어진다.
출력	몇일 뒤에 고양이들이 모일지 출력한다.
입력예시	3 7 9
출력예시	63

6.

설명	별찍기 문제에 재미를 붙인 현진이는 이번에는 조금 어려운 빗금 친 사각형을 만 들어 보기로 했다. $n \times n$ 사각형에서 k 간격 마다 빗금을 친 사각형을 출력하는 프로그램을 작성하시오. 예를 들어 $n = 7, k = 3$ 라면, <pre> ***** ** * * * * * * * ** ** * * * * * ***** </pre> 위와 같은 그림이 그려진다. 윗변을 기준으로 왼쪽부터 k 간격마다 ↙방향으로 빗금을 그으시오. 7 3인 경우, <pre> *** ** </pre> 위 빨간색부터 빗금을 그으면 된다.
입력	정사각형의 크기 n 과 빗금 간격 k 가 입력된다. ($1 \leq n, k \leq 99$)
출력	빗금 친 사각형을 출력한다.
입력예시	7 3
출력예시	<pre> ***** ** * * * * * * * ** ** * * * * * ***** </pre>

7.

설명	<p>희승이는 생명과학에 관심이 생겨 왕개미를 연구하고 있었다.</p> <p>왕개미를 유심히 살펴보던 중 특별히 성실해 보이는 개미가 있었는데, 그 개미는 개미굴에서 나와 먹이까지 가장 빠른 길로 이동하는 것이었다. 개미는 오른쪽으로 움직이다가 벽을 만나면 아래쪽으로 움직여 가장 빠른 길로 움직였다.</p> <p>(오른쪽에 길이 나타나면 다시 오른쪽으로 움직인다.)</p> <p>이에 호기심이 생긴 희승이는 그 개미를 미로 상자에 넣고 살펴보기 시작하였다. 미로 상자에 넣은 개미는 먹이를 찾았거나, 더 이상 움직일 수 없을 때까지 오른쪽 또는 아래쪽으로만 움직였다.</p> <p>미로 상자의 구조가 0(갈 수 있는 곳), 1(벽 또는 장애물)로 주어지고, 먹이가 2로 주어질 때, 성실한 개미의 이동 경로를 예상해보자.</p> <p>단, 맨 아래의 가장 오른쪽에 도착한 경우, 더 이상 움직일 수 없는 경우, 먹이를 찾은 경우에는</p> <p>더이상 이동하지 않고 그 곳에 머무른다고 가정한다.</p> <p>미로 상자의 테두리는 모두 벽으로 되어 있으며, 개미집은 반드시 (2, 2)에 존재하기 때문에 개미는 (2, 2)에서 출발한다.</p>
입력	10*10 크기의 미로 상자의 구조와 먹이의 위치가 입력된다.
출력	성실한 개미가 이동한 경로를 9로 표시해 출력한다.
입력예시	<pre> 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 0 1 1 0 0 0 0 1 2 1 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 </pre>
출력예시	<pre> 1 1 1 1 1 1 1 1 1 1 1 9 9 1 0 0 0 0 0 1 1 0 9 1 1 1 0 0 0 1 1 0 9 9 9 9 9 1 0 1 1 0 0 0 0 0 9 1 0 1 1 0 0 0 0 1 9 1 0 1 1 0 0 0 0 1 9 1 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 </pre>

1주일간의 평가는 이정도로 하겠습니다. 위 문제들을 푸시고 어려웠던 부분들은 각 파트를 더 정비하시기 바랍니다. 이제부터 진도가 쏟아지듯이 빨라지기 때문에 지금까지 배운 기초부분들은 탄탄하게 잡고 가는 것이 중요합니다.

다음 문제들은 항상 매일 풀던 문제들입니다.

1. 스도쿠를 풀어주는 프로그램을 만들어보자!

a.

스도쿠를 풀기 위해, 스도쿠 푸는 로직을 while문 안에 넣는다.

그럼, 이 while문이 끝날 조건은 무엇일까?

isEnd라는 함수를 만들어보자.

isEnd함수는 만약 while문이 더 돌아도 되는 상황에서는 True를, 스도쿠가 다 끝난 상황에서는 False를 반환한다.

(만약 생각이 떠오르지 않는다면 이 힌트를 보라. board안에 있는 숫자들 중 0이 하나라도 있다면 while문은 계속 돌아야 한다.)

b.

스도쿠를 풀기 위해서는, possibles에 있는 가능성들을 지워나가야 할 것이다.

이 가능성을 지우는 removePossibles라는 함수를 만들어보자.

이 함수는 b_list와 p_list 총 두 개를 인수로 받는다.

b_list는 가로, 세로 또는 박스, 현재 보드에서 9개의 숫자들을 1차원 List로 저장한 것이다.

p_list는 b_list에 해당하는 칸에 남은 가능한 수들을 가져온 2차원 List이다.

가령 다음과 같은 상황이 있다고 해보자.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

첫 번째 줄을 연산하는 경우,

b_list = [5, 3, 0, 0, 7, 0, 0, 0, 0]

p_list = [[5], [3], [1, 2, 3, 4, 5, 6, 7, 8, 9], [1, 2, 3, 4, 5, 6, 7, 8, 9], [7], [1, 2, 3, 4, 5, 6, 7, 8, 9], [1, 2, 3, 4, 5, 6, 7, 8, 9], [1, 2, 3, 4, 5, 6, 7, 8, 9], [1, 2, 3, 4, 5, 6, 7, 8, 9]]

와 같은 형태일 것이다. (p_list는 진행 과정에 따라 달라질 수 있다.)

위 두 리스트를 받아서, p_list에서 불가능한 경우를 지우고, 연산된 p_list를 반환하자.

위 예시와 같은 상황에서, 이 함수에서 반환되는 p_list는 다음과 같을 것이다.

p_list = [[5], [3], [1, 2, 4, 6, 8, 9], [1, 2, 4, 6, 8, 9], [7], [1, 2, 4, 6, 8, 9], [1, 2, 4, 6, 8, 9], [1, 2, 4, 6, 8, 9], [1, 2, 4, 6, 8, 9]]