

1.

먼저 함수 내에서 n 의 값이 짝수인지 아닌지 검사 후 `nlIsEvenNum` 이라는 변수에 해당 논리값을 넣어주었습니다. 그 후 문제에서 제시되어있는 재귀적 정의대로 함수의 리턴값을 설정해줍니다.

$n = 0$ 인 경우 1을, n 이 짝수인 경우와 홀수인 경우 또한 각각 재귀적으로 함수를 호출하는 코드를 작성하였습니다.

위 코드를 실행한 결과는 아래와 같습니다.

```
>> Problem_1

ans =

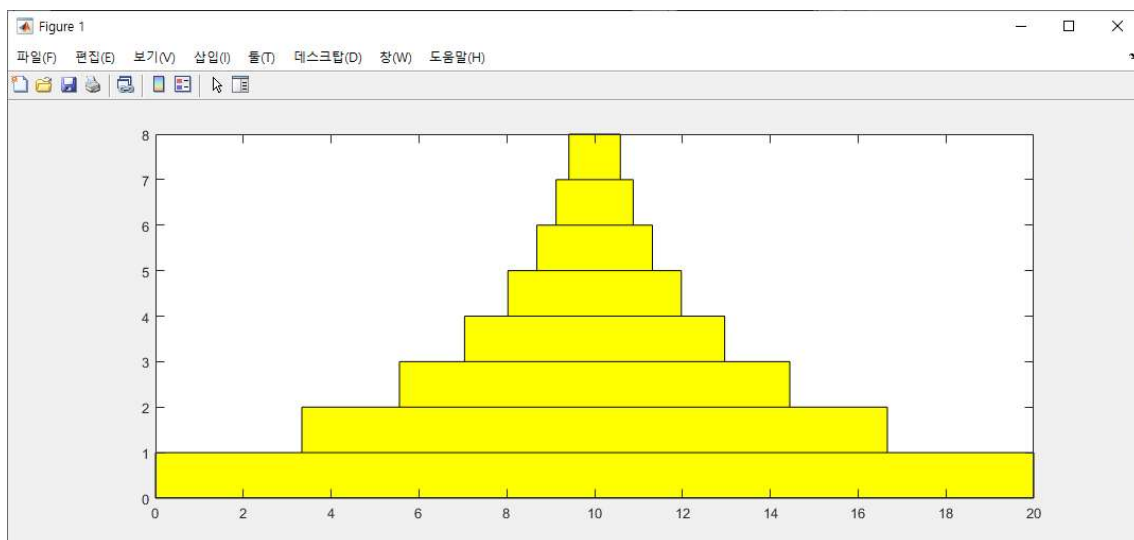
    81
```

2.

처음에 W 와 H 값을 받는 코드를 작성 한 후 직사각형을 그릴 기준이 될 좌표값인 a , b 값을 0으로 초기화해놓습니다. 그 후 `while` 반복문을 실행하는데, 조건은 $W > H$ 입니다. `while` 문이 동작하면서 바로 직사각형 하나를 먼저 그려줍니다. 그 뒤 W 값을 $2/3$ 만큼 줄여줍니다. 그 다음 원래 W 값에서 $2/3$ 배 된 W 값을 뺀 뒤 2로 나누어 주면 a 값의 x 증가량이 나오게 됩니다. 그 값을 a 에 더해줌으로써 a 좌표를 갱신해줍니다. 그 뒤 b 는 직사각형의 높이인 H 만큼 더해줍니다.

이 반복문을 실행시키게 되면 반복문 초기에 먼저 직사각형을 그려주었기 때문에 항상 $W > H$ 을 만족시키는 사각형이 그려지게 됩니다.

$L = 20$, $H = 1$ 인 경우 총 8개의 직사각형이 그려지며 결과는 아래와 같습니다.



3.

우선, 코드를 간결하기 위해 함수 두 개를 만들어주었습니다. DrawRect() 함수는 기존 위의 문제에서 사용한 코드를 사용하였고, RandomColor() 라는 'r', 'g', 'b', 'y' 중 한 문자를 같은 확률로 반환하는 함수를 만들었습니다.

다음 PartitionRect() 함수입니다. 기본단계에서 $L == 0$ 로 설정해 놓으면 $L == 1$ 일 때 문제의 예시인 초록색 정사각형 하나만 그려지는 그림이 나오지 않고, $L == 2$ 일 때의 그림이 나오는 현상이 있어 $L == 1$ 로 기본단계를 수정하였습니다.

각 기본단계와 재귀단계는 가로형, 세로형 직사각형일 때를 각각 구분합니다. 기본 단계에서, 가로형 직사각형인 경우, a, b 기준 좌표로부터 넘겨받은 height 길이만큼의 정사각형을 색칠합니다. 그 후 나머지 공간에 흰색으로 색을 칠합니다. 흰색을 칠하는 경우는 재귀단계에서는 필요하지 않으므로 마지막 단계인 기본단계에서만 칠해줍니다. 그리고 세로형 직사각형인 경우에는 a, b 기준 좌표로부터 넘겨받은 base의 길이만큼 정사각형을 칠합니다. 그 후 나머지 공간에 흰색으로 색칠해줍니다.

재귀단계에서, 가로형 직사각형인 경우, 먼저 기준좌표인 a, b로부터 height 길이만큼의 정사각형을 색칠해 줍니다. 그 후 남은 사각형의 정보들을 가지고 재귀호출합니다. a의 좌표는 height 길이만큼 늘린 $a + height$ 가 되고, b는 변하지 않습니다. 그리고 base는 height만큼 줄어 $base - height$ 가 되며, height의 길이는 변하지 않습니다.

세로형 직사각형인 경우, 먼저 기준좌표인 a, b로부터 base 길이만큼의 정사각형을 색칠해 줍니다. 그 후 남은 사각형의 정보들을 가지고 재귀호출합니다. a의 좌표는 변하지 않고, b의 좌표는 base 길이 만큼 늘린 $b + base$ 가 됩니다. base는 그대로 변하지 않고, height의 길이는 base만큼 줄어 $height - base$ 가 됩니다.

