You find yourself in Bananaland trying to buy a banana. You are super rich so you have an unlimited supply of banana-coins, but you are trying to use as few coins as possible.

The coin values available in Bananaland are stored in a sorted array coins. coins[0] = 1, and for each i (0 < i < coins.length)coins[i] is divisible by coins[i - 1]. Find the minimal number of banana-coins you'll have to spend to buy a banana given the banana's price.

**Example**

For coins = [1, 2, 10] and price = 28, the output should be  
minimalNumberOfCoins(coins, price) = 6.

You have to use 10 twice, and 2 four times.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] array.integer coins**

The coin values available in Bananaland.

*Constraints:*  
1 ≤ coins.length ≤ 5,  
1 ≤ coins[i] ≤ 120.

* **[input] integer price**

A positive integer representing the price of the banana.

*Constraints:*  
8 ≤ price ≤ 250.

* **[output] integer**

The minimal number of coins you can use to buy the banana.

<https://codefights.com/arcade/code-arcade/well-of-integration/sGwCfM5FzX7LhLcdk>

static int minimalNumberOfCoins(int[] coins, int price)

{

int i = coins.Length - 1;

int sum = 0, ans =0;

while (i >= 0)

{

while (sum + coins[i] <= price)

{

sum += coins[i];

ans++;

}

i--;

}

return ans;

}