322. Coin Change

You are given an integer array `coins` representing coins of different denominations and an integer `amount` representing a total amount of money.

Return *the fewest number of coins that you need to make up that amount*. If that amount of money cannot be made up by any combination of the coins, return `-1`.

You may assume that you have an infinite number of each kind of coin.

Example 1:

```
Input: coins = [1,2,5], amount = 11
Output: 3
Explanation: 11 = 5 + 5 + 1
```

Example 2:

```
Input: coins = [2], amount = 3
Output: -1
```

Example 3:

```
Input: coins = [1], amount = 0
Output: 0
```

Constraints:

- $1 \leq n \leq 12$
- $1 \leq coins_i \leq 2^{31} - 1$
- $0 \leq amount \leq 10^4$

Overnight

Your task is to calculate how many ways you can get the sum n by rolling the dice. Each roll of the dice produces a result between 1 and 6.

For example, if n = 3, the options are:

- 1 + 1 + 1
- 1 + 2
- 2 + 1
- 3

Input

The input is a number, n: target amount.

Printout

Your program should print one integer: how many ways can you get the sum.

Bounds

- $1 \leq n \leq 50$

Example

Input:

```
3
```

Printout:

```
4
```