1151: Number of Dice Rolls with Target Sum

You have 'n' rolls and each die has 'K' faces numbered 1-K

Given `n`, 'K' and `target` return possible # of ways to roll the dice so that the sum of face up numbers equals the target.

Since answer may be large return % 10^9+7

Input: n=1 K=6 target=3
Output: 1

Input: n=2 K=6 target=7

Output: 6

1+6 2+5 3+4 4+3 5+2 6+1

Approach: Brute Force

```
std::vector<int> dice(n, 1);
```

We can check all possible solutions.

Could speed up w/ map with dice values as key.

1 1
1 2
1 3
1 4
1 5
1 6

2 1
2 2
3 2

1 1 1
1 1 2

1.) Create a data structure to store n, K, target called DS

2.) Create global map< DS, int> to store already calculated num rolls.

3.) Recursively call num rolls.

If value in map, return answer

★ Base case is there is only one answer if n = 1 , 1 or 0. This is our exit condition

★ If The highest die value multiplied by the total # of dice is still less than target, there are zero possible ways.

★ Since minimum is 1 , if target value is less than total dice used, there are zero possible ways.

★ If n, k, t exist in map, return value.

For each possible dice value augment assign

$f(n-1, K, target - possible\ value)$

This simulates each individual dice roll.