Q1

$2x_1 + 4x_2 + 5x_3 + 8x_4 = 10,\ x_i : is\ non\ negative\ integer$

we set the coefficient of x_i as a_i. and RHS as S.(S=10)

note that $\frac{(x_1+x_2+x_3+x_4)!}{(x_1)!(x_2)!(x_3)!(x_4)!}$ is the permutation of 4 unique objects and x_i is the number of object i.

so in order to explain the problem better we can say that Given a value $S = 10$ and we have given a set of 4 cards with values $A = a_1, \ldots, a_4$, we must find the total number of arrangements of cards that will make sum S. we can use each card as much as we want.

so we mapped the coefficient a_i with a card with value a_i, x_i as the number of cards a_i we use, and $\frac{(x_1+x_2+x_3+x_4)!}{(x_1)!(x_2)!(x_3)!(x_4)!}$ as the number of card arrengments.

solution

we first initalize a DP array with length $10 + 1$. DP_i represents the number of ways we can arrenge cards with total value of i.

we initialize DP_0 as 1 (the only way to have 0 sum is to select no cards). for each $1 \leq i \leq 10$, we count the ways of permuting cards with total amount i.

for each $1 \leq i \leq 10$:

1. for each cards a_j:

- if $a_j > i$: we can not use this card becous a_j is already bigger than the sum of all cards i. so $DP_{i_{new}} = DP_i$
- if $a_j \leq i$: when we choose a_j the sum is $i - a_j$. so we need to find the total permutations of the cards that add up to $i - a_j$ which is DP_{i-a_j} and has been calculated. we need to add the number of these new permutations to the old ones, so $DP_{i_{new}} = DP_i + DP_{i-a_j}$

```
def all_permutation(rhs,coefs):
    dp=[0 for i in range(rhs + 1)]
    dp[0]=1
    for i in range(1,rhs + 1):
        for j in range(len(coefs)):
            if (coefs[j] <= i):
                dp[i] += dp[i - coefs[j]]

    return dp[rhs]
```

```
rhs=10
```

```
coefs=[2,4,5,8]
sum=all_permutation(rhs,coefs)
print("sum: ",sum)
```

```
sum:  11
```