

题目描述：

某组织举行会议，来了多个代表团同时到达，接待处只有一辆汽车，可以同时接待多个代表团，为了提高车辆利用率，请帮接待员计算可以坐满车的接待方案，输出方案数量。

约束：

1、一个团只能上一辆车，并且代表团人数（代表团数量小于 30，每个代表团人数小于 30）
小于汽车容量（汽车容量小于 100）

2、需要将车辆坐满

输入描述：

第一行 代表团人数，英文逗号隔开，代表团数量小于 30，每个代表团人数小于 30

第二行 汽车载客量，汽车容量小于 100

输出描述：

坐满汽车的方案数量

如果无解输出 0

补充说明：

各代表团人数 5,4,2,3,2,4,9

汽车载客量 10

输出 4

解释 以下几种方式都可以坐满车，所以，优先接待输出为 4

[2, 3, 5]

[2, 4, 4]

[2, 3, 5]

[2, 4, 4]

示例 1

输入:

5, 4, 2, 3, 2, 4, 9

10

输出:

4

说明:

解释 以下几种方式都可以坐满车, 所以, 优先接待输出为 4

[2, 3, 5]

[2, 4, 4]

[2, 3, 5]

[2, 4, 4]

```
import java.util.*;
```

```
// 注意类名必须为 Main, 不要有任何 package xxx 信息
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner in = new Scanner(System.in);
```

```
        // 注意 hasNext 和 hasNextLine 的区别
```

```
        while (in.hasNext()) { // 注意 while 处理多个 case
```

```
            //背包问题
```

```
            String[] strs = in.nextLine().split(",");
```

```
            int[] arr = new int[strs.length];
```

```
            for(int i=0;i<strs.length;i++){
```

```
                try {
```

```
                    arr[i]=Integer.parseInt(strs[i]);
```

```
                } catch (NumberFormatException e) {
```

```
                    System.out.print(0);
```

```
                    return;
```

```
                }
```

```
            }
```

```
            int target = 0;
```

```
            try {
```

```
                target = in.nextInt();
```

```
            } catch (Exception e) {
```

```
                System.out.print(0);
```

```
                return;
```

```
            }
```

```

int[] dp= new int[target+1];
dp[0]=1;
for(int num:arr){
    int[] temp = new int[target+1];
    for(int i=0;i<=target;i++){
        if(dp[i]==0) continue;
        int sum = i+num;
        if(sum<=target){
            temp[sum] += dp[i];
        }
    }
    for (int i = 0; i <= target; i++) {
        dp[i] += temp[i];
    }
}
System.out.print(dp[target]);
}
}
}

```