

# Yatzy (40%)

## Problem ID: finalp1

In this project, you implement a program for playing a very simple version of the game Yatzy. This version of the game has the following functionality:

- There is only one player.
- There are five dice, each with numbers in the range 1 to 6.
- The rolling of the dice is simulated by entering a line containing five integer values. If an empty line is entered, the program quits running.
- The player scores points for certain dice combinations. The points  $p$  are calculated as follows:
  1. **Yatzy**: All 5 dice show the same number. Then,  $p = 50$ .
  2. **Three of a kind**: The same result appears on 3 dice. Let  $v$  be the value that appears on three dice. Then,  $p = 3 \cdot v$ . Note that it is not necessary that the other dice be different from these three, but **Three of a kind** is however only applicable if the dice do not show Yatzy.
  3. **Pair**: The same result appears on 2 dice. Let  $v$  be the highest value that appears on a pair of dice. Then,  $p = 2 \cdot v$ . In other words, if two pairs occur on the dice, the one with higher points is selected. A pair is only applicable if **Three of a kind** does not appear on the dice. **Three of a kind** should always be selected in favour of a **Pair**, even if the latter gives higher points.
  4. If none of the above applies, that is, if all the dice show different values, then  $p = 0$ .
- Each roll of the dice is scored independently, regardless of what combinations have been achieved before (see Sample 2 below).

Do take a look at the provided starter code. It includes a given function `get_counts()` that counts how often each value appears in a given roll of the dice. You **can** use this function if you want.

You are **not** allowed to use any import statement in your solution, except `import typing`.

## Input

As you can see in the samples below, each line in the input contains 5 space-separated numbers. Those represent the results of rolling 5 dice. The input then ends with a single blank line indicating the end of input. This blank line can not be seen in the sample boxes, but it is there nonetheless.

## Formal input specifications

To be specific, the input consists of  $n + 1$  lines. The last line is empty, but the first  $n$  lines, call them  $l_1, l_2, \dots, l_n$ , contain dice results.

That is, for each  $j \in \{1, 2, \dots, n\}$ , the line  $l_j$  consists of a sequence  $v_j = (v_{j1}, v_{j2}, v_{j3}, v_{j4}, v_{j5})$  of 5 integers denoting dice values. The values are separated with a single space.

## Test case restrictions

In the test cases,  $n$  will be restricted to  $0 \leq n \leq 20$ , and the dice values will be integers restricted to  $1 \leq v_{ji} \leq 6$  for each  $j \in \{1, \dots, n\}$  and  $i \in \{1, \dots, 5\}$ .

**Note:** Your program is not supposed to validate this input, or refuse other input. This is just for your information about the input in the test cases. It is good if your program can handle other input as well, but you do not need to expect input that does not meet these restrictions.

## Output

For each roll of the 5 dice, the output consists of one line containing the points earned as stated above.

### Formal output specifications

To be specific, the output should consist of  $n$  lines, call them  $\lambda_1, \dots, \lambda_n$ .

For each  $j \in \{1, \dots, n\}$ , the line  $\lambda_j$  should contain one integer  $p_j$ , which should be the number of points the player scored for the dice roll in line  $l_j$  of the input.

#### Sample Input 1

3	4	2	5	4
5	1	2	5	5

#### Sample Output 1

8
15

#### Sample Input 2

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

#### Sample Output 2

3
6
50
0
18