WEEK 2 PROBLEMS

Problem 1 :

Santa has become stranded at the edge of the Solar System while delivering presents to other planets! The Elves quickly load you into a spacecraft and prepare to launch.
At the first Go / No Go poll, every Elf is Go until the Fuel Counter-Upper. They haven't determined the amount of fuel required yet.
Fuel required to launch a given module is based on its mass. Specifically, ***to find the fuel required for a module, take its mass, divide by three, round down, and subtract 2.***
For example:
For a mass of 12, divide by 3 and round down to get 4, then subtract 2 to get 2.
For a mass of 14, dividing by 3 and rounding down still yields 4, so the fuel required is also 2.
For a mass of 1969, the fuel required is 654.
For a mass of 100756, the fuel required is 33583.
The Fuel Counter-Upper needs to know the total fuel requirement. To find it, individually calculate the fuel needed for the mass of each module (your puzzle input), then add together all the fuel values.
What is the sum of the fuel requirements for all of the modules on your spacecraft?

Problem 2 :

Something is wrong with global snow production, and you've been selected to take a look. You try to ask why they can't just use a weather machine ("not powerful enough") and where they're even sending you ("the sky") and why your map looks mostly blank ("you sure ask a lot of questions") and hang on did you just say the sky ("of course, where do you think snow comes from") when you realize that the Elves are already loading you into a trebuchet ("please hold still, we need to strap you in").
As they're making the final adjustments, they discover that their calibration document (your puzzle input) has been amended by a very young Elf who was

apparently just excited to show off her art skills. Consequently, the Elves are having trouble reading the values on the document.
The newly-improved calibration document consists of lines of text; each line originally contained a specific calibration value that the Elves now need to recover. On each line, **the calibration value can be found by combining the first digit and the last digit (in that order) to form a single two-digit number**.

For example:
1abc2
pqr3stu8vwx
a1b2c3d4e5f
treb7uchet

In this example, the calibration values of these four lines are 12, 38, 15, and 77. Adding these together produces 142.
Consider your entire calibration document. What is the sum of all of the calibration values?

Problem 3 :

Table: Person

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| id          | int     |
| email       | varchar |
+-------------+---------+
```

id is the primary key (column with unique values) for this table.
Each row of this table contains an email. The emails will not contain uppercase letters.
Write a solution to delete all duplicate emails, keeping only one unique email with the smallest id.
For SQL users, please note that you are supposed to write a DELETE statement and not a SELECT one.
For Pandas users, please note that you are supposed to modify Person in place.

After running your script, the answer shown is the Person table. The driver will first compile and run your piece of code and then show the Person table. The final order of the Person table does not matter.

Example :
Input:
Person table:

```
+----+------------------+
| id | email            |
+----+------------------+
| 1  | john@example.com |
| 2  | bob@example.com  |
| 3  | john@example.com |
+----+------------------+
```

Output:

```
+----+------------------+
| id | email            |
+----+------------------+
| 1  | john@example.com |
| 2  | bob@example.com  |
+----+------------------+
```

Explanation: john@example.com is repeated two times. We keep the row with the smallest Id = 1.

Problem 4 :

You have observations of n + m 6-sided dice rolls with each face numbered from 1 to 6. n of the observations went missing, and you only have the observations of m rolls. Fortunately, you have also calculated the average value of the n + m rolls.
You are given an integer array rolls of length m where rolls[i] is the value of the ith observation. You are also given the two integers mean and n.
Return an array of length n containing the missing observations such that the average value of the n + m rolls is exactly mean. If there are multiple valid answers, return any of them. If no such array exists, return an empty array.

The average value of a set of k numbers is the sum of the numbers divided by k. Note that mean is an integer, so the sum of the n + m rolls should be divisible by n + m.

Example 1:
Input: rolls = [3,2,4,3], mean = 4, n = 2
Output: [6,6]
Explanation: The mean of all n + m rolls is (3 + 2 + 4 + 3 + 6 + 6) / 6 = 4.

Example 2:
Input: rolls = [1,5,6], mean = 3, n = 4
Output: [2,3,2,2]
Explanation: The mean of all n + m rolls is (1 + 5 + 6 + 2 + 3 + 2 + 2) / 7 = 3.

Example 3:
Input: rolls = [1,2,3,4], mean = 6, n = 4
Output: []
Explanation: It is impossible for the mean to be 6 no matter what the 4 missing rolls are.

Constraints:
m == rolls.length
1 <= n, m <= 105
1 <= rolls[i], mean <= 6
