1. How will you tackle the challenge above?

To tackle this challenge, I would write a program that reads the input CSV file and parses each cell's value to determine

Ans--Answer 1: First of all, I will if the input is present and check the values then I will return the value,

however if the input has not contained number , then I will check if the values used in the formula are present in the sheet or not,

and if the values in the formula are present then I will calculate the formula and i will return the calculated value as output.

2. What type of errors you would you check for?

When parsing and evaluating the formulas in the CSV file, I would check for the following errors:

Ans-: Formulas can sometimes result in error values when we are returning the values.

for this problem-

Division by zero errors: If a formula divides by zero, the program should return an error and skip calculating the value for that cell.

CSV format errors: Check if the input CSV file is in the correct format and contains valid data

I have to check for the no integer values. If the values are present I will use the formula,

then I will check the errors which are written below

-> Is the formula calculating the LOG of a negative number?

-> Is the formula dividing by zero?

-> Is the formula trying to calculate the special characters instead of actual values?

-> Is the formula calculating the square root of a negative number?

-> Is the formula calculating a value larger than the maximum value of the current type?

3. How might a user break your code?

Check if the output CSV file can be created and written to without errors

Users might break the code by providing an input CSV file that is not in the correct format, contains invalid data, or has formulas with errors.  
  
Ans: 'Break code' means presenting the output in a desirable form so the user may try to break our code if any error can occur and if the exceptions are not handled properly. There are some scenarios where users might break our code. The scenarios can be:

-> If the code is not tested properly.

-> If the code is not mandated regularly.

-> If the code is not properly version-controlled.

-> If the code does not report errors.

Therefore, we need to implement error handling and input validation mechanisms to prevent these scenarios.