1) Create a program that simulates a **text data compression system**. The user enters a paragraph as a **string**.
   The program should:
   a. Count the frequency of each word and store it in a **dictionary**.
   b. Convert the dictionary into a **list of tuples**, where each tuple is (word, frequency).
   c. Sort this list by frequency in descending order.
   d. Allow the user to retrieve the frequency of a particular word.

   Exception handling requirements:
   e. Handle **KeyError** if the word is not present.
   f. Handle **IndexError** if the user tries to access an invalid index in the tuple list.
   g. Handle **AttributeError** if the user enters a non-string input.
2) Create a program to manage employee records and calculate monthly payroll using **tuples, dictionaries, strings, and exception handling**.
   a. Store employees in a dictionary, where keys are employee IDs (strings) and values are tuples containing (name, base_salary, department).
   b. Allow the user to search for an employee by entering their ID.
   c. Handle the following cases with exceptions:
      i. If the employee ID does not exist, handle the KeyError with a proper message.
      ii. If the user accidentally enters a non-numeric salary adjustment (bonus or deduction), catch the ValueError.
      iii. If the adjustment results in a negative final salary, raise and handle a **custom exception** NegativeSalaryError.
   d. After updating, print the employee's final salary slip in a well-formatted string.
   e. The program should run in a loop until the user types "exit".