

Problem 1

Viết chương trình tính toán giá trị xấp xỉ của hàm e^x dựa theo công thức sau đây:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

Lưu ý: Các bạn có thể tự cài đặt hàm giai thừa riêng bên ngoài và gán lại cho biến cần sử dụng để code gọn.

Problem 2

Viết chương trình đếm số lượng chữ xuất hiện trong một từ, với key là chữ cái và value là số lần xuất hiện

1. **Input:** một từ
2. **Output:** Dictionary đếm số lần xuất hiện các chữ xuất hiện
3. **Note:** Giả sử các từ nhập vào đều có các chữ cái thuộc [a-z] hoặc [A-Z]

Code listings 1: Đây là các ví dụ các bạn không cần thiết đặt tên giống ví dụ

```
# Examples 1(a)
string = "Happiness"
count_chars ( string )
>> {"H": 1 , "a": 1 , "e": 1 , "i": 1 , "n": 1 , "p": 2 , "s": 2}

string = "smiles"
count_chars ( string )
{"e": 1 , "i": 1 , "l": 1 , "m": 1 , "s": 2}
```

Problem 3

Cho một chuỗi `coded_str` chứa các ký tự đã được mã hóa theo cách sau: $k[\text{string}]$, trong đó:

1. k là số lần lặp lại của `string` bên trong dấu [].
2. `coded_str` chỉ chứa các ký tự tiếng Anh (a-z, A-Z). Không có ký tự đặc biệt.
3. Nếu có số, chúng chỉ xuất hiện ở k và phải đứng trước [].
4. $k \in [1, 9]$.

Note: k trong range [1, 9]

Ví dụ

1. **Input:** `coded_str = "2[abc]3[cd]ef"`
2. **Output:** `"abccabccdcddcdef"`
3. **Input:** `coded_str = "abc3[cd]xyz"`
4. **Output:** `"abccdcddcdxyz"`

Problem 4

Cho danh sách `data` gồm các chuỗi ký tự. Mỗi chuỗi có dạng: `key1=value1;key2=value2;...`
Trong đó:

1. **key** là tên một thuộc tính (chuỗi không chứa ký tự `=` hoặc `;`).
2. **value** có thể là một số nguyên, số thực, hoặc chuỗi ký tự.

Yêu cầu:

1. Phân tích dữ liệu thành một dictionary lớn, tạo một dictionary từ danh sách trên, trong đó:
 - (a) Mỗi **key** từ chuỗi trở thành một **key** trong dictionary.
 - (b) Giá trị tương ứng là danh sách tất cả các **value** của **key** đó xuất hiện trong danh sách `data`.
2. Với mỗi **key**, tính:
 - (a) Số lượng giá trị duy nhất (**unique_count**).
 - (b) Giá trị lớn nhất (**max_value**, nếu là số).
 - (c) Độ dài chuỗi lớn nhất (**max_length**, nếu là chuỗi).
3. Lưu kết quả dưới dạng một dictionary mới.

Ví dụ:

Code listings 2: Đây là các ví dụ các bạn không cần thiết đặt tên giống ví dụ

```
Input :
data = [
    "name=Alice ; age=30; score=85.5 " ,
```

```
"name=Bob;age=25;score=90" ,
"name=Alice;age=30;score=92" ,
"city=NewYork;name=Eve;age=35;score=88" ,
"city=London;name=Alice;age=30;score=85.5"
]
Output:
dict = {
    "name": ["Alice", "Bob", "Alice", "Eve", "Alice"],
    "age": [30, 25, 30, 35, 30],
    "score": [85.5, 90, 92, 88, 85.5],
    "city": ["NewYork", "London"]
}
result = {
    "name": {"unique_count": 3, "max_value": None, "max_length": 5},
    "age": {"unique_count": 3, "max_value": 35, "max_length": None},
    "score": {"unique_count": 4, "max_value": 92, "max_length": None},
    "city": {"unique_count": 2, "max_value": None, "max_length": 7}
}
```