Midterm examination

Sinh viên tra mã đề theo mã số sinh viên trong bảng dưới đây và tìm nội dung đề tương ứng ở các trang kế tiếp.

MSSV	Mã đề		MSSV	Mã đề		MSSV	Mã đề		MSSV	Mã đề
1712258	1		19120190	1		19120492	1		19120619	1
1712345	2		19120207	2		19120494	2		19120633	2
1712376	3		19120220	3		19120501	3		19120673	3
1712401	4		19120241	4		19120505	4		19120683	4
1712688	5		19120252	5		19120522	5		19120684	5
18120181	6		19120301	6		19120528	6		19120688	6
18120216	7		19120315	7		19120534	7		19120691	7
18120384	8		19120318	8		19120539	8		19120702	8
18120413	1		19120330	1		19120549	1		19120707	1
18120462	2		19120331	2		19120551	2		19120709	2
19120057	3		19120364	3		19120557	3		19120716	3
19120124	4		19120412	4		19120562	4		19120724	4
19120142	5		19120454	5		19120568	5		19120729	5
19120145	6		19120462	6		19120572	6			
19120186	7		19120485	7		19120602	7			
19120189	8		19120489	8		19120615	8			

Mã đề 01

Input: Cho ngữ liệu gồm nhiều văn bản. Nội dung của mỗi văn bản chỉ chứa ký tự viết thường, khoảng trắng và ký tự xuống dòng.

Ví dụ

Tập tin	file01	file02	file03
Nội dung	full of water full of sweet	dessert of summer treat to eat	treat to enjoy oh so cool

Yêu cầu: Xác định số lần xuất hiện của các chuỗi hai từ liên tiếp (2-grams)

Output: Danh sách (*2-gram*, *frequency*), với *2-gram* là chuỗi hai từ liên tiếp có trong ngữ liệu và *frequency* là số lần xuất hiện của *2-gram*. Không quan trọng thứ tự hiển thị của các bộ giá trị.

Ví dụ:

Nội dung tập tin đầu ra
full of 2
of water 1
of sweet 1
dessert of 1
of summer 1
treat to 2
to eat 1
to enjoy 1
oh so 1
so cool 1

Lưu ý

- Sinh viên có thể cần phải thay đổi tham số và kiểu dữ liệu của tham số trong hàm.

Tài liệu hỗ trợ: Tập tin set01.zip, gồm có

- Mã nguồn WordCount.java cài đặt chương trình WordCount chuẩn
- Tập tin đầu vào mẫu

Mã đề 02

Input: Cho ngữ liệu gồm nhiều văn bản. Nội dung của mỗi văn bản chỉ chứa ký tự viết thường, khoảng trắng và ký tự xuống dòng.

Ví dụ

Tập tin	file01	file02	file03
Nội dung	yellow sun yellow sun sunshine down so bright	the sun is bright yellow warm sunshine	the sun so warm sunshine sunshine

Yêu cầu: Xác định số lần xuất hiện trung bình qua các dòng cho mỗi từ phân biệt trong ngữ liệu

- Chỉ xét trên các dòng mà từ có xuất hiện.
- Ví dụ: từ "abc" xuất hiện 1 lần ở dòng 2, 0 lần ở dòng 2, và 3 lần ở dòng 3 thì số lần xuất hiện trung bình là (1+3)/2 = 2.

Output: Danh sách (*word*, *average*), với *word* là từ phân biệt có trong ngữ liệu và *average* là số lần xuất hiện trung bình qua các dòng của *word*. Không quan trọng thứ tự hiển thị của các bộ giá trị.

Nội dung tập tin đầu ra
yellow 1.5
sun 1.333
sunshine 1.333
down 1
so 1
bright 1
the 1
is 1
warm 1

Lưu ý

- Sinh viên có thể cần phải thay đổi tham số và kiểu dữ liệu của tham số trong hàm.

Tài liệu hỗ trợ: Tập tin set01.zip, gồm có

- Mã nguồn WordCount.java cài đặt chương trình WordCount chuẩn
- Tập tin đầu vào mẫu

Mã đề 03

Input: Cho ngữ liệu gồm nhiều văn bản. Nội dung của mỗi văn bản chỉ chứa ký tự viết thường, khoảng trắng và ký tự xuống dòng.

Ví dụ

Tập tin	file01	file02	file03
Nội dung	yellow sun yellow sun sunshine down so bright	the sun is bright yellow warm sunshine	the sun so warm sunshine sunshine

Yêu cầu: Với mỗi từ phân biệt trong ngữ liệu, in ra số lần xuất hiện lớn nhất và số lần xuất hiện nhỏ nhất qua các dòng.

- Chỉ xét trên các dòng mà từ có xuất hiện.
- Số lần xuất hiện lớn nhất có thể bằng số lần xuất hiện nhỏ nhất
- Ví dụ: từ abc xuất hiện 1 lần ở dòng 2, 2 lần ở dòng 2, và 3 lần ở dòng 3 thì hai giá trị cần xác định là 3 và 1.

Output: Danh sách (*word*, *max, min*), với *word* là từ phân biệt có trong ngữ liệu, *max* và *min* là số lần xuất hiện lớn nhất và nhỏ nhất qua các dòng của *word*. Không quan trọng thứ tự hiển thị của các bộ giá trị.

Nội dung tập tin đầu ra
yellow 2 1
sun 2 1
sunshine 2 1
down 1 1
so 1 1
bright 1 1
the 1 1
is 1 1
warm 1 1

Lưu ý

- Sinh viên có thể cần phải thay đổi tham số và kiểu dữ liệu của tham số trong hàm.

Tài liệu hỗ trợ: Tập tin set01.zip, gồm có

- Mã nguồn WordCount.java cài đặt chương trình WordCount chuẩn
- Tập tin đầu vào mẫu

Mã đề 04

Input: Cho ngữ liệu gồm nhiều văn bản. Nội dung của mỗi văn bản chỉ chứa ký tự viết thường, khoảng trắng và ký tự xuống dòng.

Ví dụ

Tập tin	file01	file02	file03
Nội dung	yellow sun yellow sun sunshine down so bright	the sun is bright yellow warm sunshine	the sun so warm sunshine sunshine

Yêu cầu: Xác định số lượng từ phân biệt có một ký tự, hai ký tự, ba ký tự, v.v., cho đến n ký tự (với n là số ký tự của từ dài nhất).

- Không cần tạo bộ giá trị cho trường hợp không có từ nào thỏa

Output: Danh sách (*numchar*, *numword*), với *numchar* là số lượng ký tự và *numword* là số lượng từ phân biệt có *numchar* ký tự. Không quan trọng thứ tự hiển thị của các bộ giá trị.

Nội dung tập tin đầu ra
2 2
3 2
4 2
6 2
8 1

Lưu ý

- Sinh viên có thể cần phải thay đổi tham số và kiểu dữ liệu của tham số trong hàm.

Tài liệu hỗ trợ: Tập tin set01.zip, gồm có

- Mã nguồn WordCount.java cài đặt chương trình WordCount chuẩn
- Tập tin đầu vào mẫu

Mã đề 05

Input: Cho ngữ liệu gồm nhiều văn bản. Nội dung của mỗi văn bản chỉ chứa ký tự viết thường, khoảng trắng và ký tự xuống dòng.

Ví dụ

Tập tin	file01	file02	file03
Nội dung	yellow sun yellow sun sunshine down so bright	the sun is bright yellow warm sunshine	the sun so warm sunshine sunshine

Yêu cầu: Xác định số lượng dòng mà mỗi từ phân biệt trong ngữ liệu có xuất hiện.

Output: Danh sách (*word*, *numline*), với *word* là từ phân biệt có trong ngữ liệu và *numline* là số lượng dòng có chứa *word*. Không quan trọng thứ tự hiển thị của các bộ giá trị.

Nội dung tập tin đầu ra
yellow 2
sun 3
sunshine 3
down 1
so 2
bright 2
the 2
is 1
warm 1

Lưu ý

- Sinh viên có thể cần phải thay đổi tham số và kiểu dữ liệu của tham số trong hàm.

Tài liệu hỗ trợ: Tập tin set01.zip, gồm có

- Mã nguồn WordCount.java cài đặt chương trình WordCount chuẩn
- Tập tin đầu vào mẫu

Mã đề 06

Input: Cho ngữ liệu gồm nhiều văn bản. Mỗi văn bản được tổ chức thành nhiều dòng, và mỗi dòng tuân theo định dạng key-value như sau

⟨key⟩ ⟨value⟩

trong đó ⟨key⟩ là chuỗi (không khoảng trắng) dùng để phân biệt văn bản và ⟨value⟩ là chuỗi chỉ chứa ký tự viết thường và khoảng trắng. ⟨key⟩ và ⟨value⟩ phân cách nhau bằng dấu tab.

Ví dụ

Tập tin	file01	file02	file03
Nội dung	id1 full of water id1 full of sweet	id2 dessert of summer id2 treat to eat	id3 treat to enjoy id3 oh so cool

Yêu cầu: Xác định số lượng văn bản chứa mỗi từ phân biệt trong ngữ liệu

Output: Danh sách (*word*, *numdoc*), với *word* là từ phân biệt có trong ngữ liệu và *numdoc* là số lượng văn bản có chứa *word*. Không quan trọng thứ tự hiển thị của các bộ giá trị.

Nội dung tập tin đầu ra
full 1
of 2
water 1
sweet 1
summer 1
treat 2
to 2
……

Lưu ý

- Sinh viên có thể cần phải thay đổi tham số và kiểu dữ liệu của tham số trong hàm.
- Tham khảo kiểu KeyValueTextInputFormat: hoạt động tương tự TextInputFormat, tức là xử lý theo dòng, nhưng mỗi dòng có dạng key-value cách nhau bằng dấu tab

Tài liệu hỗ trợ: Tập tin set02.zip, gồm có

- Mã nguồn WordCount-KeyValue.java mô phỏng chương trình WordCount nhưng đã cấu hình để nhận dữ liệu đầu vào dạng KeyValueTextInputFormat
- Tập tin đầu vào mẫu

Mã đề 07

Input: Cho ngữ liệu gồm nhiều văn bản. Mỗi văn bản được tổ chức thành nhiều dòng, và mỗi dòng tuân theo định dạng key-value như sau

⟨key⟩ ⟨value⟩

trong đó ⟨key⟩ là chuỗi (không khoảng trắng) dùng để phân biệt văn bản và ⟨value⟩ là chuỗi chỉ chứa ký tự viết thường và khoảng trắng. ⟨key⟩ và ⟨value⟩ phân cách nhau bằng dấu tab.

Ví dụ

Tập tin	file01	file02	file03
Nội dung	id1 full of water id1 full of sweet	id2 dessert of summer id2 treat to eat	id3 treat to enjoy id3 oh so cool

Yêu cầu: Với mỗi từ trong ngữ liệu, xác định văn bản mà trong đó từ xuất hiện nhiều lần nhất

- Nếu có nhiều văn bản thỏa điều kiện thì chỉ cần xuất ra một

Output: Danh sách (*word*, *doc*), với *word* là từ phân biệt có trong ngữ liệu và *doc* là mã văn bản mà *word* xuất hiện nhiều lần nhất. Không quan trọng thứ tự hiển thị của các bộ giá trị.

Nội dung tập tin đầu ra
full id1
of id1
water id1
sweet id1
dessert id2
……

Lưu ý

- Sinh viên có thể cần phải thay đổi tham số và kiểu dữ liệu của tham số trong hàm.
- Tham khảo kiểu KeyValueTextInputFormat: hoạt động tương tự TextInputFormat, tức là xử lý theo dòng, nhưng mỗi dòng có dạng key-value cách nhau bằng dấu tab

Tài liệu hỗ trợ: Tập tin set02.zip, gồm có

- Mã nguồn WordCount-KeyValue.java mô phỏng chương trình WordCount nhưng đã cấu hình để nhận dữ liệu đầu vào dạng KeyValueTextInputFormat
- Tập tin đầu vào mẫu

Mã đề 08

Input: Cho ngữ liệu gồm nhiều văn bản. Mỗi văn bản được tổ chức thành nhiều dòng, và mỗi dòng tuân theo định dạng key-value như sau

⟨key⟩ ⟨value⟩

trong đó ⟨key⟩ là chuỗi (không khoảng trắng) dùng để phân biệt văn bản và ⟨value⟩ là chuỗi chỉ chứa ký tự viết thường và khoảng trắng. ⟨key⟩ và ⟨value⟩ phân cách nhau bằng dấu tab.

Ví dụ

Tập tin	file01	file02	file03
Nội dung	id1 full of water id1 full of sweet	id2 dessert of summer id2 treat to eat	id3 treat to enjoy id3 oh so cool

Yêu cầu: Xác định số lượng từ phân biệt chứa trong mỗi văn bản

Output: Danh sách (*word*, *count*), với *word* là từ phân biệt có trong ngữ liệu và *count* là số lượng từ phân biệt chứa trong văn bản. Không quan trọng thứ tự hiển thị của các bộ giá trị.

Nội dung tập tin đầu ra
id1 4 id2 6 id3 6

Lưu ý

- Sinh viên có thể cần phải thay đổi tham số và kiểu dữ liệu của tham số trong hàm.
- Tham khảo kiểu KeyValueTextInputFormat: hoạt động tương tự TextInputFormat, tức là xử lý theo dòng, nhưng mỗi dòng có dạng key-value cách nhau bằng dấu tab

Tài liệu hỗ trợ: Tập tin set02.zip, gồm có

- Mã nguồn WordCount-KeyValue.java mô phỏng chương trình WordCount nhưng đã cấu hình để nhận dữ liệu đầu vào dạng KeyValueTextInputFormat
- Tập tin đầu vào mẫu