Assignment 5 – Testing the Word Term Frequencies Calculating Program

RunLog:

- Environment: Windows 10 OS | Run on March 1st 2018 | 11:00PM || Python pyc file was decompiled and ran as py file
- As of March 2nd 2018
 - o BlackBoxTesting
 - Total Bugs found : 5 bugs (Test ID: 2, 5, 6, 9, 11)
 - Total Issues found: 1 issue (Test ID: 3)
 - o Stress Testing
 - Total Issues found: 2 Issues (Test ID:)
 - o Combinational Testing
 - o Internationalization Testing
- Please refer the table on the next page. Bugs are red coded

Test ID Execution Order Type	Summary	Input	Expected Output	Procedure	Test Type	Actual Output
1 correct	Summary information using input file	mike.txt containing "mike was here"	3 tokens 3 unique tokens	1. Execute the command "python tf.pyc -s -i mike.txt" 2. Validate output is correct.	Functional Validation	
						Vivek-Pc@kev MINGW64 /e/Marist/Semester2/SoftwareVerificationAndMaintenan ce/termFrequency_2_ \$ python termFrequency.py -v Please enter a file name: input.json Debug mode! Input file: input.json
						Whole Input File Content: தமிழ்
						=======================================
						Output file: ./termFreqOutput.txt
2	Verbose Mode:	input.json file	"Can process only English language statements" since I assume this program	1. Execute the command "python termFrequency.py -v" then "input.json" 2. Validate output is having info stating "Non-	Functional and Internationaliz	Term Frequencies: "多口": {"count": "1", "tokenIds": ["0"]} Traceback (most recent call last): File "termFrequency.py", line 170, in <module> output(i, wordKey[0]) File "termFrequency.py", line 41, in output outputFile.write(""%s": {"count": "%s", "tokenIds": %s}' % (wordKey, len(sentenceDict[wordKey]), str(sentenceDict[wordKey]).replace(""", "")))</module>
Incorrect -	-Used other language, say Tamil language in input file	containing "தமிழ்"	does not support different languages	English language	ation Validation	File "C:\DEV\PYTHON3\lib\encodings\cp1252.py", line 19, in encode

						return
						codecs.charmap_encode(input,self.errors,encoding_table)[
						0]
						UnicodeEncodeError: 'charmap' codec can't encode
						characters in position 1-2: character maps to <undefined></undefined>
						Vivek-Pc@kev MINGW64
						/e/Marist/Semester2/SoftwareVerificationAndMaintenan
						ce/termFrequency_2_
						\$ python termFrequency.py -v
						Please enter a file name: input.json
						Debug mode!
						Input file: input.json
				1. Execute the		Whole Input File Content:
				command "python termFrequency.py		
				-v" then		
				"input.json"		
3				2. Validate output		Output file: ./termFreqOutput.txt
Feature			Information stating like	is having info		
Request	Verbose Mode:	Input.json file	"No input found in the	stating "File	Functional	T
Issue	-Used empty file	contains nothing	input.json file"	contains no input"	Testing	Term Frequencies:
				1. Execute the command "python		
		Input.json file		termFrequency.py		
		contains the		-v" then		Used factorial python program inside the file and the
4		python		"input.ison"		output came with no issue. This is because, the pgm
Correct	Verbose Mode:	programing	X tokens with X unique	2. Validate that	Functional	converts all the inputs inside the input ison file as string and
Output	-Used Code inside the file	code	tokens	output is correct	Testing	then process them ignoring the symbols
						Vivek-Pc@kev MINGW64
						/e/Marist/Semester2/SoftwareVerificationAndMaintenan
		Input.json file				ce/termFrequency_2_
		contains				\$ python termFrequency.py -v
	Verbose Mode:	"hello hello		1. Execute the		Please enter a file name: input.json
	-Used combination of symbols	hello hello		command "python		Debug mode!
	to verify the consistency of	hai-hai hai hai		termFrequency.py		Input file: input.json
5	ignorance. The "_" is not being	abc***abc abc		-v" then		
Incorrect	parsed well	abc	X tokens with X unique	"input.json"		
output	- The "_" should also be	pop(pop) pop	tokens should be	2. Validate that	Functional	Whole Input File Content: hello_hello hello hello
Bug	parsed like other symbols.	pop"	obtained	the output	Testing	hai-hai hai

						abc***abc abc abc
						pop(pop) pop pop
						ρορ(ρορ) ρορ ρορ
						==============
						Output file: ./termFreqOutput.txt
						Term Frequencies:
						"hai": {"count": "4", "tokenIds": ["3", "4", "5", "6"]}
						"hai": {"count": "4", "tokenlds": ["3", "4", "5", "6"]} "abc": {"count": "4", "tokenlds": ["7", "8", "9", "10"]} "pop": {"count": "4", "tokenlds": ["11", "12", "13", "14"]}
						"pop": {"count": "4", "tokenIds": ["11", "12", "13", "14"]}
						"hello": {"count": "2", "tokenIds": ["1", "2"]}
						"hello_hello": {"count": "1", "tokenlds": ["0"]}
						Vivek-Pc@kev MINGW64
						/e/Marist/Semester2/SoftwareVerificationAndMaintenan
						ce/termFrequency_2_
						\$ python termFrequency.py -v
						Please enter a file name: input.json
						Debug mode! Input file: input.json
						input tile: input.json
						Whole Input File Content:
						vvnoie input the Comeni.
		Input.json file				
		contains "				
						~~~
						~~~~
						········· ^^^^^^^ \$\$\$
						=======================================
	Verbose Mode:					
	-Used symbols to verify the			1. Execute the		
	consistency of ignorance of it.			command "python		Output file: ./termFreqOutput.txt
1	- The "" symbol used	~~~~		termFrequency.py		
6	inside the input ison file	~~~~		-v" then		
Incorrect	- Those are still symbols and	^^^^^^	N I	"input.json"		Term Frequencies:
output	they are not parsed and		No token output should	2. Validate the	Functional	"": {"count": "2", "tokenIds": ["4", "8"]} "": {"count": "2", "tokenIds": ["5", "6"]}
Bug	ignored.	\$\$\$"	be obtained	output	Testing	: { count : "2", "tokenIds": ["5", "6"]}

	-					"": {"count": "1", "tokenIds":
						["O"]}
						"": {"count": "1", "tokenlds": ["1"]}
						"": {"count": "1", "tokenIds": ["2"]} "": {"count": "1", "tokenIds": ["3"]}
						: { count : 1 , lokelinds : [3]} ": {"count": "1", "tokenIds": ["7"]}
						Vivek-Pc@kev MINGW64
						/e/Marist/Semester2/SoftwareVerificationAndMaintenan
						ce/termFrequency_2_
						\$ python termFrequency.py -v
						Please enter a file name: input.json
						Debug mode!
						Input file: input.json
		Input.json file				
		contains "[][][]				Whole Input File Content: [][][][]
		"""" ;;;;; ;;;; ::::				
		/// ~~~,,,,,				
		/// ~~~````				
						!! !! @@ @@ ## \$\$ % ^ & * () +=
				1. Execute the command "python		===========
				termFrequency.py		
	Verbose Mode:			-v" then		Output file: ./termFreqOutput.txt
	-Used symbol combinations	!! !!		"input.json"		
7	again to verify the ignorance	@@ @@ ##		2. Validate that		
correct	of it.	\$\$ % ^ & *() -	No tokens should be	the output should	Functional	Term Frequencies:
output		+="	obtained	contain nothing	Testing	"_": {"count": "1", "tokenIds": ["0"]}
		Innut tyt		1. Execute the		Vivek-Pc@kev MINGW64
		Input.txt		command "python termFrequency.py		/e/Marist/Semester2/SoftwareVerificationAndMaintenan ce/termFrequency_2_
		contains		-v" then		\$ python termFrequency.py -o term.txt
8	Output file mode usage -o	3,880,800		"input.json" and		Time Before inputting the fileName: 2018-03-02
correct	-Used the input file that	words with		validate the	Functional	12:41:22.169234
output	contains few bigger values	50,409 lines	X tokens	output	Testing	Please enter a file name: input.txt

				2. Verify that the given output file is		Time Before completing the file write function: 2018-03-02 12:41:38.581536
				obtained with no		Time After completing the file write function: 2018-03-02 12:41:38.586522
9 Incorrect Output Bug	Output file mode usage with other options -d -s -Used the output file mode along with other options like -d or -s or both -d -s along with - o while executing the command, the given output file not created - For example, Positive case: refer 8 above \$ python termFrequency.py -o output.txt -When above command used, the output.txt file is created with no issues. Negative case: \$ python termFrequency.py -o -d -s output.txt -Now, when this command executed, the output.txt file is not created.	Input.txt contains "Hello hai hello world hai hello"	X tokens	1. Execute the command "python termFrequency.py -v" then "input.json" and validate the output	Functional Testing	Vivek-Pc@kev MINGW64 /e/Marist/Semester2/SoftwareVerificationAndMaintenan ce/termFrequency_2_ \$ python termFrequency.py -o -d -s output.txt Time Before inputting the fileName: 2018-03-02 12:52:58.017764 Please enter a file name: input.txt Time Before completing the file write function: 2018-03- 02 12:53:00.341114 Time After completing the file write function: 2018-03-02 12:53:00.346119 Summary: Tokens: 9 Unique Tokens: 6 Max Fequency: ('hello', 3) Min Fequency: 1 Vivek-Pc@kev MINGW64
10	Verbose mode with -a sorting feature testing			1. Execute the command "python termFrequency.py -v" then "input.json" and validate the		/e/Marist/Semester2/SoftwareVerificationAndMaintenan ce/termFrequency_2_ \$ python termFrequency.py -v Time Before inputting the fileName: 2018-03-02 12:57:01.796715 Please enter a file name: input.txt Debug mode! Input file: input.txt Whole Input File Content: Hello Ada Joey Ada Hello ===================================
Correct	-Used the input.txt file	Input.txt		output	Functional	
Code	containing the unsorted words	contains "	X tokens		Tesging	Output file: ./termFreqOutput.txt

		1		ı	I .
					Term Frequencies:
					"hello": {"count": "2", "tokenIds": ["0", "4"]}
					"ada": {"count": "2", "tokenlds": ["1", "3"]}
					"joey": {"count": "1", "tokenlds": ["2"]}
					Time Before completing the file write function: 2018-03-
					02 12:57:03.437732
					Time After completing the file write function: 2018-03-02
					12:57:03.442211
					12:57:03:442211
					Vivek-Pc@kev MINGW64
					/e/Marist/Semester2/SoftwareVerificationAndMaintenan
					ce/termFrequency_2_
					\$ python termFrequency.py -v -a
					Time Before inputting the fileName: 2018-03-02
					12:57:32.105676
					Please enter a file name: input.txt
					Debug mode!
					Input file: input.txt
					Whole Input File Content: Hello Ada Joey Ada Hello
					=======================================
					Output file: ./termFreqOutput.txt
					Term Frequencies:
					"ada": {"count": "2", "tokenIds": ["1", "3"]}
					"hello": {"count": "2", "tokenIds": ["0", "4"]}
					"joey": {"count": "1", "tokenIds": ["2"]}
					Time Before completing the file write function: 2018-03-
					02 12:57:36.949347
					Time After completing the file write function: 2018-03-02
					12:57:36.954351
11			1. Execute the		
Incorrect	Input Mode	Inputting in the	command "python		Vivek-Pc@kev MINGW64
	-Used the input mode and	command	termFrequency.py	Functional	/e/Marist/Semester2/SoftwareVerificationAndMaintenan
Output	-osed me input mode and				

- The i	input is being received	"Hello world	"input.json" and	\$ python termFrequency.py -n
again	n and again	hello world	validate the	Hello world hello world
- There	re should be a way to	still accepting	output	still accepting the input
stop a	accepting and process the	the input		not valid!!!
input o	given.	not valid!!!"		Traceback (most recent call last):
				File "termFrequency.py", line 125, in <module></module>
				content = sys.stdin.read()
				KeyboardInterrupt

	STRESS TESTING									
Test ID	Execution Order Type	Summary of what I did	Input	Expected Output	Procedure	Actual Output Obtained: (trimmed output a little to be view legibly)				
						Vivek-Pc@kev MINGW64 /e/Marist/Semester2/SoftwareVerificationAndMaintenance/termFreq uency_2_ \$ python termFrequency.py -v Please enter a file name: input.txt Debug mode! Input file: input.txt				
1	1 Correct Output	Normal file with normal inputs	Input.txt contains one line	X tokens	Execute the command "python termFrequency.py -v" then "input.json" and validate the output	Whole Input File Content: "hello world hai hello wolod world hello hello"				

						Output file: ./termFreqOutput.txt Term Frequencies: "hello": {"count": "4", "tokenIds": ["0", "3", "6", "7"]} "world": {"count": "2", "tokenIds": ["1", "5"]} "hai": {"count": "1", "tokenIds": ["2"]}
						"wolod": {"count": "1", "tokenIds": ["4"]} Time Before completing the file write function: 2018-03-02 12:06:06.435161 Time After completing the file write function: 2018-03-02 12:06:06.436153
						Vivek-Pc@kev MINGW64 /e/Marist/Semester2/SoftwareVerificationAndMaintenance/termFreq uency_2_ \$ python termFrequency.py -v Please enter a file name: input.txt Debug mode! Input file: input.txt
	2		Input.txt		1. Execute the command "python termFrequency.py -v" then	Whole Input File Content: Stress testing is a type of performance testing focused on determining an application's robustness, availability, and reliability under extreme conditions. The goal of stress testing is to identify application issues that arise or become apparent only under extreme conditions. These conditions can include heavy loads, high concurrency, or limited computational resources. Proper stress testing is useful in finding synchronization and timing bugs, interlock problems, priority problems, and resource loss bugs. The idea is to stress a system to the breaking point in order to find bugs that will make that break potentially harmful. The system is not expected
2	Correct Output	Small file testing	contains 122 words	X tokens	"input.json" and validate the output	to process the overload without adequate resources, but to behave (e.g., fail) in an acceptable manner (e.g., not corrupting or losing data).

	1	1	1	1	1	
						=======================================
						Output file: ./termFreqOutput.txt
						Time Before completing the file write function: 2018-03-02 12:18:31.055440 Time After completing the file write function: 2018-03-02 12:18:31.067437
3	3 Correct	Big file testing	Input.txt contains 122,000 words	X tokens	Execute the command "python termFrequency.py -v" then "input.json" and validate the output	Time Before completing the file write function: 2018-03-02 12:21:47.030644 Time After completing the file write function: 2018-03-02 12:21:47.035623
4	4 Correct	Bigger file testing	Input.txt contains 3,880,800 words with 50,409 lines	X tokens	Execute the command "python termFrequency.py -v" then "input.json" and validate the output	Time Before inputting the fileName: 2018-03-02 12:38:41.372525 Time Before completing the file write function: 2018-03-02 12:39:23.800657 Time After completing the file write function: 2018-03-02 12:39:23.805682
5	5 Memory Error – I guess Issue is with my system	Bigger file testing	Input.txt contains 17,791,500 words with 147,918 lines	X tokens	Execute the command "python termFrequency.py -v" then "input.json" and validate the output	Traceback (most recent call last): File "termFrequency.py", line 140, in <module> tokens = re.findall('[\\w]+', content.lower().replace('\n', ' ').replace('\r', ' ').replace(', ', ' ').replace(', ', ' ')) MemoryError</module>
6	6 Memory Error – I guess Issue is	Bigger file testing	Input.txt contains 24,766,000 words with	X tokens	Execute the command "python termFrequency.py -v" then "input.json" and validate the output	Traceback (most recent call last): File "termFrequency.py", line 140, in <module> tokens = re.findall('[\\w]+', content.lower().replace('\n', ' ').replace('\r', ' ').replace(', ', ' ').replace(', ', ' '))</module>

with my system	203,204 lines		MemoryError