

CISC 327: Course Project Assignment #6

Team name: Runtime Terror

Team members: Anna Chulukov, 17avc2@queensu.ca, 20082947;
Faranak Sharifi, 17fsb@queensu.ca, 20068900;
Marwan ElKhodary, m.elkhodary@queensu.ca, 20022212;
Ryan Fernandes, 17rf@queensu.ca, 20067569

Question 1a: Source listing of Daily script implemented in bash:

```
1  #!/usr/bin/env bash
2
3  #runs the program using inputs in text file
4  python3 frontend.py < $1
5  python3 frontend.py < $2
6  python3 frontend.py < $3
7
8  #runs the program to merge the transactionSummaryFile
9  python3 daily.py
10
11 #runs backend
12 python3 backend.py
```

Question 1b:Source listing of Daily script helper in Python, iterates through all transaction summary files and merges them:

```

1  import glob # used to retrieve all transaction summary files in a list from the directory the program is in
2  import functools # used to flatten a 2d array
3  import operator # used to flatten a 2d array
4  import os
5
6
7  def readFiles():
8      transactionCodesNotFlat = []
9      for f in glob.glob("your_transaction_summary_*"): # iterates through all files within the directory that start
10         # with 'your_transaction_summary_'
11         file = open(str(f), "r")
12         transactionCodesNotFlat.append(file.read().splitlines()) # array of transaction codes is initially a 2d array
13         os.remove(f)
14         # due to this line
15         transactionCodes = functools.reduce(operator.iconcat, transactionCodesNotFlat, []) # flattens 2d array into a 1d
16         # array
17
18     return transactionCodes
19
20
21 def mergeTransactionSummaryFiles(transactionCodes):
22     with open("merged_transaction_summary.txt", "w") as file:
23         for transaction in transactionCodes:
24             file.write(transaction + "\n")
25
26
27 def main():
28     transactionsList = readFiles()
29     mergeTransactionSummaryFiles(transactionsList)
30
31
32 main()

```

Question 2:

Daily script is initially run with Input1.txt, Input2.txt and Input3.txt

Input1.txt	Input2.txt	Input3.txt
login	login	login
agent	agent	agent
createacct	createacct	createacct
1234567	1111111	2345678
Marwan	Anya	Ryan
no	no	no

Question 3:

Printout of the Merged Transaction Summary File from the same Daily run in Question 2.

1	NEW	<u>2345678</u>	000	0000000	Ryan
2	EOS	0000000	000	0000000	***
3	NEW	1234567	000	0000000	Marwan
4	EOS	0000000	000	0000000	***
5	NEW	1111111	000	0000000	Anya
6	EOS	0000000	000	0000000	***

Question 4:

Daily run 1:

1	1111111	<u>30000</u>	Anya
2	1234567	30000	Marwan
3	2345678	30000	Ryan

Daily run 2:

1	1111111	<u>29500</u>	Anya
2	1234567	29500	Marwan
3	2345678	29500	Ryan

Daily run 3:

1	1111111	<u>29000</u>	Anya
2	1234567	30500	Marwan
3	2345678	29000	Ryan

Daily run 4:

1	1111111	29000	Anya
2	1234567	30500	Marwan
3	1245874	000	John
4	2345678	29000	Ryan
5	5468136	000	Sarah
6	5629873	000	Dove

Daily run 5:

1	1111111	29000	Anya
2	1234567	30500	Marwan
3	1245874	000	John
4	2345678	29000	Ryan
5	5468136	12000	Sarah
6	5629873	45000	Dove

Question 5:

Integration Defect Report
<p><u>Front-End:</u> Changes made the frontend.py</p> <ul style="list-style-type: none"> * Within logout(), changed opening a file in the directory to simply creating a file without using command line arguments. Similarly readFile() was changed to remove command line arguments and have the program open the validaccountslist within the program * Within logout, we have changed the opening of a transaction summary file, so that every time a user logs out, the program creates a transaction summary file with the date/time in the name * Within createacct(), changed the storing of transaction codes to properly mimic assignment descriptions
<p><u>Back-End:</u> Changes made to backend.py</p> <ul style="list-style-type: none"> * Changed all instances of using command line arguments within the program, such as opening the merged transaction summary file, by hard-coding it into the program * Changed readTransFile() so that once all the transactions are stored in an array, the merged transaction summary file is cleared of its contents to be used the next day

Note:

```
./daily.sh: line 5: Input4.txt0: No such file or directory
./daily.sh: line 6: Input4.txt1: No such file or directory
./daily.sh: line 7: Input4.txt2: No such file or directory
./daily.sh: line 5: Input4.txt3: No such file or directory
./daily.sh: line 6: Input4.txt4: No such file or directory
./daily.sh: line 7: Input4.txt5: No such file or directory
```

Since we had a problem running the weekly scripts with all the given arguments at once [Input4.txt - Input18.txt, error provided above. After debugging we realized that it was because a shell script can only take nine arguments, after that there needs to be {} brackets around the arguments.