

DATA WAREHOUSING AND DATA MINING LAB (CSD-421) LAB ASSIGNMENT 3

AKSHAT RAJ VANSI (185520)

JANUARY 27, 2022



Contents

1	Program 1	2
1.1	Question	2
1.2	Code	2
1.3	Output	3
2	Program 2	4
2.1	Question	4
2.2	Code	4
2.3	Output	5

1 Program 1

1.1 Question

Create 500 txt files in a directory. Every file contains 20,000 lines and every line contain random string of length 20 characters.

1.2 Code

```
1  __author__ = 'Akshat Raj Vansh'
2  __version__ = '0.1.0'
3  __license__ = 'MIT'
4
5  import random
6  import string
7  import os
8
9
10 def generateRandomLines():
11     text = []
12     for i in range(0, 20000):
13         text.append(''.join(random.choices(
14             string.ascii_letters+string.digits, k=20))+'\n')
15     return text
16
17
18 def writeToFile(filename):
19     file = open(filename.encode('unicode_escape'), 'w+')
20     randomString = generateRandomLines()
21     file.writelines(randomString)
22     file.close()
23
24
25 def main():
26     directory = os.getcwd()
27     directory = directory[:directory.rfind("\\")]
28     filename = directory+'\\Files\\textfile'
29     for i in range(1, 501):
30         writeToFile(filename+str(i)+'.txt')
31
32
33 if __name__ == '__main__':
34     main()
```

1.3 Output

Question 1 Outputs



2 Program 2

2.1 Question

Calculate the execution time to convert all the file to upper case. Save the results in csv file as given below.

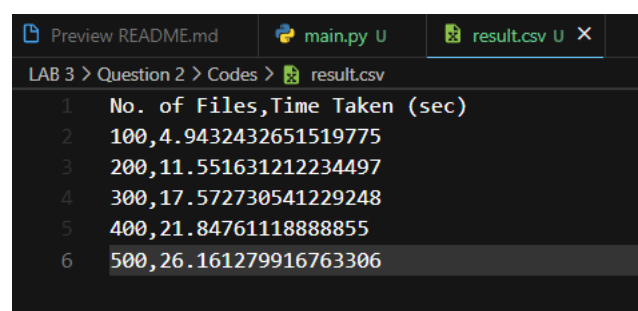
2.2 Code

```
1  __author__ = 'Akshat Raj Vansh'
2  __version__ = '0.1.0'
3  __license__ = 'MIT'
4
5  import time
6  import os
7  import random
8  import string
9
10
11 def encodeString(word):
12     return word.encode('unicode_escape')
13
14
15 def updateFile(filename):
16     file = open(encodeString(filename), 'r')
17     originalText = file.read()
18     file.close()
19     file = open(encodeString(filename), 'w+')
20     originalText = originalText.splitlines(keepends=True)
21     updatedText = [x.upper() for x in originalText]
22     file.writelines(updatedText)
23     file.close()
24
25
26 def generateRandomLines():
27     text = []
28     for i in range(0, 20000):
29         text.append(''.join(random.choices(
30             string.ascii_letters+string.digits, k=20))+'\n')
31     return text
32
33
34 def writeToFile(filename):
35     file = open(filename.encode('unicode_escape'), 'w+')
36     randomString = generateRandomLines()
37     file.writelines(randomString)
38     file.close()
39
40
41 def rollback():
42     directory = os.getcwd()
43     directory = directory[:directory.rfind("\\")]
44     filename = directory+'\\Files\\textfile'
45     for i in range(1, 501):
46         writeToFile(filename+str(i)+'.txt')
47
48
49 def writeResults(result):
```

```
50     file = open('result.csv', 'w+')
51     row_out = []
52     res_out = []
53     for row in result:
54         row_out.append(','.join(x for x in row))
55     for res in row_out:
56         res_out.append('\n'.join(x for x in row_out))
57     file.writelines(res_out)
58     file.close()
59
60 def main():
61     directory = os.getcwd()
62     directory = directory[:directory.rfind("\\")]
63     filename = directory+'\\Files\\textfile'
64
65     table = [['No. of Files', 'Time Taken (sec)']]
66     for j in range(1, 6):
67         rollback()
68         begin = time.time()
69         for i in range(1, j * 100 + 1):
70             updateFile(filename+str(i)+'.txt')
71         end = time.time()
72         table.append([str(j*100), str(end-begin)])
73         print('Finished updating {n} files in {t} seconds'.format(n=j*100, t= end-begin))
74     writeResults(table)
75
76
77 if __name__ == '__main__':
78     main()
```

2.3 Output

Question 2 Outputs



	No. of Files,Time Taken (sec)
1	100,4.9432432651519775
2	200,11.551631212234497
3	300,17.572730541229248
4	400,21.84761118888855
5	500,26.161279916763306