

CSc11300 Programming Languages Makeup Midterm Exam

April 20, 2021

Instructor: Ahmet C. Yuksel

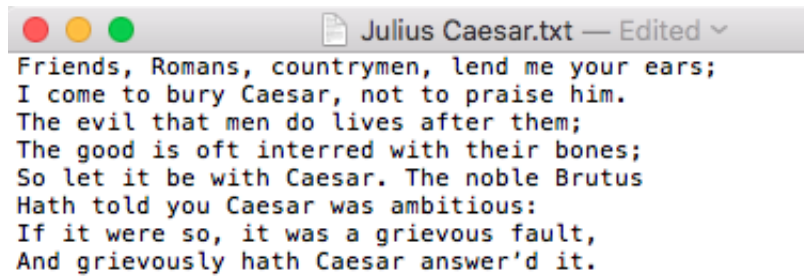
Deadline: April 22, 11:59 pm

Use Python3 to solve the programming problems. No online modules, codes, functions are allowed. There is a strict deadline, sending it later will face heavy penalty. You may not use the stuff that we have not learned yet. Any signs of collaboration will result in 0 grade.

1. (15 points) Let A and B two text files, containing some binary lines between of length 1..N. Where N is the line with the maximum number of characters, 1 the minimum number of characters in each line.

Design a Python3 function to compare each line of the files, character by character, and if there is a match of the characters, create a variable representing the similarity between the lines of each file, and return that similarity variable measures the similarity between two files.

2. (10 points) Use a Python3 compound statement to open the sample text file below. Write a program uses a function to exchange the first n words in the second line with the last n words in the last line and write the change back to the same file.(n is a variable, your function should work with any non-empty text file)



```
Friends, Romans, countrymen, lend me your ears;  
I come to bury Caesar, not to praise him.  
The evil that men do lives after them;  
The good is oft interred with their bones;  
So let it be with Caesar. The noble Brutus  
Hath told you Caesar was ambitious:  
If it were so, it was a grievous fault,  
And grievously hath Caesar answer'd it.
```

3. (10 points) Let $I = [1, 2.09, 3, 4, 5, 6, 'a', 8, 9, 10.14, 11, 'b']$ be a Python3 list. And let $J = \{1, 2, 7\}$ be a set of indexes. Write a Python3 function to create a 2D list out of I where each row contains 2 elements and then design another function to return the projection of 2D I onto J.(Should work with any 1D list I and J created with respect to the size of I)

4. (15 points) Read the contents of a text file, then place the contents to a 2D list as seen the figure below, where 1, is the element at `list[0][0]` is the first line of the file and 12, (`list[2][3]`) is the 12th line of the file. Write a Python3 function to randomly shuffle the elements of 2D list and write it back to the file with that particular order. For randomly selecting an index from the 2d list, you can use `random.randint(a,b)` or `random.shuffle(list)` function from the random module, where the `randint` function returns an integer N and $a \leq N \leq b$. and the `shuffle` function returns a shuffled list.

		col →			
row ↓		0	1	2	3
	0	1	2	3	4
	1	5	6	7	8
	2	9	10	11	12

5. (15 points) Write the Python3 code to give the following output without using `'''` (triple quotation marks). You are allowed to use `Print()` function just once.

```

**      81 \quest
*      82
*      83
****    84 \newli
****    85 \newli
****    86 \inclu
*****  87 \vspac
*****  88
*****  89
*****  90
*****
*****
*****

```

6. (15 points) Write the Python3 code to give the following output without using `'''` (triple quotation marks). You are allowed to use `Print()` function just once.

```

*      67
**     68 \vspac
***    69 \newpa
****   70 \quest
****   index
****   functi
***** return
***** 71 \newli
***** 72 \newli
***** 73 \inclu
***** 74 \newli
***** 75 \newli
***** 76 For in
***** the 53
***** sizes
***** 77 When 5
***** size 1
***** 78 5J_13=
***** 79 5J_23=
***** 80 5J_33=
***** \quest
*****
]

```

7. (10 points) Design a Python3 function to apply caesar shifting to any text file. In this encrypting method, we shift the characters by some variable n . If the current character is 'a' and $n=2$, then we shift a to 'c', (2 characters) and write it to the file/same-different does not matter. We will not encrypt whitespace/special characters (such as punctuation marks), but we will keep them in the text file between words. The text file is not case sensitive. If n is a greater number than the number of characters of the English alphabet, it should go back to the alphabet, if for a, $n=28$ then 'c'. For the values of n we must be able to use negative numbers as well, in that case it will go backwards to determine the character for the encryption.

An example:

T='aaaaabbbcc', $n=2$

T='ccccdddee'

8. (10 points) Design a Python3 function to compare every prefix of a string X to every element of string Y, if there is a match, place it in a python set, sort and reversely sort the set, and return the sorted and reversely sorted sets. Good luck.