

THEORY QUESTIONS ASSIGNMENT

Python based theory

To be completed at student's own pace and submitted before given deadline

NO	TASK	POINTS
PYTHON		
1	Theory questions	30
2	String methods	29
3	List methods	11
4	Dictionary methods	11
5	Tuple methods	2
6	Set methods	12
7	File methods	5
TOTAL		100

1. Python theory questions	30 points
-----------------------------------	------------------

1. What is Python and what are its main features?

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics intended to be relatively straightforward for humans to read and write and for computers to read and process. Its high-level built in data structures, combined with dynamic typing and dynamic binding.

2. Discuss the difference between Python 2 and Python 3

Python 2 and python 3 have semantic and syntactic differences like the function print in python 3 is necessary the round brackets in python 2 you don't need them, In Python 2, an implicit str type is ASCII. But in Python 3 implicit str type is Unicode, In python 2 exists the xrange method but python 3 doesn't have it. Python 2 has the method iteritems() for dictionaries but Python 3 don't. The input in python 2 enters the value as its type if it is "int" the input the value of variable is "int" but in python 3 the value no matter its type became string.

3. What is PEP 8?

PEP means Python Enterprise Proposal, PEP 8 is a document written by Guido van Rossum, Barry Warsaw, and Nick Coghlan that contain guidelines and best practices on how to write consistent, understandable, readable and beautiful code in python.

4. **In computing / computer science what is a program?**

Program is a set of sequential instructions to get a result and realize specific functions.

5. **In computing / computer science what is a process?**

The process is the execution of a program in the computer, containing the program code and its activities. The process can be made up by multiple threads of execution.

6. **In computing / computer science what is cache?**

The cache is a hardware or software component that stores data for future use.

7. **In computing / computer science what is a thread and what do we mean by multithreading?**

Multithreading is CPU feature, it is a generalized programming and execution model that allows multiple threads to exist in the context of one process. The threads share the process's resources, but are independent in execution. Related to the times of the programming when is running correct or not.

8. **In computing / computer science what is concurrency and parallelism and what are the differences?**

Concurrency is related to an application executing and running the task and managing the multiple computations at the same time. In Parallelism the application has the task divided in small subtasks that run multiple computations simultaneously.

9. **What is GIL in Python and how does it work?**

The Global Interpreter Lock (GIL) is a single lock on the interpreter itself which adds a rule to the execution of any Python bytecode allowing only one thread to hold the control of the Python interpreter. Preventing deadlocks, multiple threads at once also prevents race conditions and ensures thread safety and doesn't introduce much performance overhead.

10. **What do these software development principles mean: DRY, KISS, BDUF**

KISS : The "Keep It Simple, Stupid" principle reminds us that the software or applications design and deployment must be done simple, without complexities, easy to understand the procedures and function. This principle is related to the code that must be easy to debug. All this is done thinking in future maintenance by the developer and other operations.

DRY : The "Don't Repeat Yourself" (also known as "Duplication Is Evil") principle indicates us that every software engineer should avoid the repetition of information or methods within their work in order because it is important to avoid the redundancy.

The system is divided in fragments, the code in smaller code or segment to have an easier execution.

BDUF: Big Design Up Front (BDUF), this software development approach tell us that the program's design is to be completed and perfected before that program's implementation is started.

11. What is a Garbage Collector in Python and how does it work?

The garbage collector tracks all the objects in memory. A new object starts its life in the first generation of the garbage collector when Python executes a garbage collection process on a generation and an object survives, it moves up into a second, older generation.

12. How is memory managed in Python?

Memory management is a process that allows that applications read and write the objects and data structures also where to put the data, to do it, involves a dynamic memory allocation by a private heap. The Python memory manager ensured the heap internally.

13. What is a Python module?

Module is the file with the ".py" extension, in the module there is a python code that contains functions to be used in a future in an application, the module can be imported for other python programs or main program.

14. What is docstring in Python?

In a simple way a docstring are the information inside the quotation marks (""" """) used to document and explain what a module, function, class, method does. The docstrings are useful because it allow us and others to understand better the code.

15. What is pickling and unpickling in Python? Example usage.

"Pickling" is related to the pickled module, it is the process whereby a Python object hierarchy is converted into a byte stream (binary file or bytes-like object) this means not human readable, and "unpickling" is the inverse operation, whereby a byte stream is converted back into an object hierarchy.

Example:

```
import pickle

America_countries={'South_america' : ['Peru', 'Argentina', 'Uruguay','Brazil' , 'Colombia', 'Ecuador',
'Chile','Paraguay','Venezuela', 'Trinidad y Tobago'], 'North_america':['Canada', 'United States', 'Mexico']}

with open('America.txt', 'wb') as America_file:
    pickle.dump(America_countries, America_file)

with open('America.txt', 'rb') as America_file:
    American_cont=pickle.load(America_file)

print(American_cont)
```

16. What are the tools that help to find bugs or perform static analysis?

Pychecker is a software development tool, it is an opensource tool for static analysis that detects the bugs from source code by examining the structural elements of the code, it can also notify about the style and complexity of the bug. It can be installed by using `pip install Pychecker`.

Pylint is a tool to that checks and control for warnings and errors. It is also opensource tool for static code analysis, it looks for programming errors, type errors and is used for coding standard. It also checks the length of each programming line, the variable names according to the project style. It can be used in python IDEs such as Pycharm, Spyder, Eclipse, and Jupyter. It can be installed by using `pip install Pylint`.

17. How are arguments passed in Python by value or by reference? Give an example.

Passed by reference; when you change what a parameter refers to within a function, the change also realize in the calling function. The value is passed into the function directly because that the argument you're passing to the function is a reference to a variable that exists in memory.

Passed by value; It means that the value is directly passed as the value to the argument of the function. The changes that we made in a copy of the variable don't modify the value of the variable outside the function.

Examples: An example can be when we have a function that add a new element to the list here it is possible see if a list is affected by the add value or not.

18. What are Dictionary and List comprehensions in Python? Provide examples.

List comprehensions and dictionary comprehensions are powerful substitute to for-loops and also lambda functions. They are effective, fast and very simple ways to do a beautiful and understandable and readable code.

Example of list comprehension:

```
numbers_n=[1,2,3,4,5,6]
newlist = [x**2 for x in numbers_n ]
print(newlist)
```

Example of dictionary comprehension:

```
numbers_n=[1,2,3,4,5,6]
new_dict = {num: num**2 for num in numbers_n}
print(new_dict)
```

19. What is namespace in Python?

A namespace is a system that it is an aggrupation of actual defined symbolic

names that reference for every object in Python along with information. It is like a dictionary (namespace) with keys (object names).

20. What is pass in Python?

The pass statement is used as a placeholder for future code when we left a empty code for a future use like the student files (google drive) of the class of this course and then we write the code in the class. If we execute a code with pass nothing happens but we avoid errors.

21. What is unit test in Python?

Unit test is used to verify a single component of code, usually modularized as a function if the functions are working according the expected, here we put the expected value and the function if they are the same our test is successful. To do this we use the module unittest.

22. In Python what is slicing?

Slice is a method that can be slice() method or obj[start:stop:step] that we use to access to elements of a list according to their index. Slicing is a useful technique.

23. What is a negative index in Python?

The negative index start from the last element of an array (example: List), if we want the -1 index this is the last element, then the -2 is the element before the last and so on.

24. How can the ternary operators be used in python? Give an example.

Ternary operators(<,>,<=,>=,==) are conditional expressions, They evaluate something based if the condition is true or false. It can be used with loops, if else conditions.

Example:

```
min = 20<15  
print(min)
```

25. What does this mean: *args, **kwargs? And why would we use it?

*args, allow us to take variable length arguments in a function, instead def sum(x,y) we could use def(*num).

kwargs, it also allow us to take variable length argument, but it is used for keyword argument. So to use it we write def poem(data):

26. How are range and xrange different from one another?

The both are similar because provide us values from a beginning to an end (specified), but the difference is the type of the result, range results in a list and xrange in an xrange object.

27. What is Flask and what can we use it for?

Flask is a web framework that provides useful tools. It is used when We want to created web applications in python and sometimes in a single file.

28. What are clustered and non-clustered index in a relational database?

Clustered Index; if the records are stored in order or in sequential order this is a clustered index. Usually the order is the value of the primary key and also it is condition to have a primary key to create this index that means the data can't have repeated values. Sometimes one clustered index is used as a primary key. Navigate quickly to the correct record doing a binary search is known as a table seek.

Non-clustered index; this index is a data structure of records that use single and multiple columns or fields as the constraint. The index store a sorted table of pointers of one column and other columns equivalent of the original table. It is very efficient and decreases query time because it moves through data much easier. The query optimizer with this index test all columns or queries that test first columns, then second columns and so on.

29. What is a 'deadlock' a relational database?

A deadlock is a complicated situation when two or more transactions are constantly waiting for one another to release and give up locks.

30. What is a 'livelock' a relational database?

A Livelock is a situation similar to a deadlock that occurs when a request for an exclusive lock is denied repeatedly.

**2. Python string methods:
describe each method and provide an example**

29 points

METHOD	DESCRIPTION	EXAMPLE
capitalize()	capitalize is a method that works with string data type. It converts the first letter of the word or phrase in capital letter.	phrase="the lord of the flies" phrase.capitalize()
casefold()	casefold is a method that converts all characters of the string into lowercase letters and returns a new string. It is similar to lower method.	name="Aldous Leonard Huxley" name.casefold()
center()	center is a method that returns a string with a center position according the length parameter.	book="Brave New World" book.center(20, "-")
count()	This method return the number of a specified character. It works with string, float and number type.	word="Brown Bear, Brown Bear, What Do You See?" word.count("Bear")
endswith()	This string method returns True or False if a string ends in a specified character.	text="A war fought between the Achaeans and the Trojans,the Trojan War" text.endswith("Trojan War")
find()	This string method find the first occurrence of a character and returns its index.	text="The city of Troy was founded on the western tip of Anatolia in the 19th century,and while it is almost certainly the Homeric Troy" text.find("Troy")
format()	This method with two arguments (value, specification of format) returns the value with new format.	print(format(12, "b"))
index()	Returns the first position or index of a specified element a list, string, number, list.	word="Brown Bear, Brown Bear, What Do You See?" word.index("Bear")
isalnum()	This method returns true if all the characters are alphanumeric or false if they aren't.	music = "Blink182" music.isalnum()

isalpha()	This method returns true if all the characters belongs to the alphabet (a-z) otherwise return false.	music = "Blink182" music.isalpha()
isdigit()	This method returns true if all the characters are digits if they aren't returns false.	password = "123456" password.isdigit()
islower()	Returns True if all the characters are in lowercase otherwise returns false.	print("The French Revolution of 1789".islower())
isnumeric()	Returns True if all the characters are numeric (0-9), returns false if they aren't.	count="123" print(count.isnumeric())
isspace()	Returns True if there are only whitespaces.	txt = " a " whitespace = txt.isspace() print(whitespace)
istitle()	Returns True if the first letter or character are uppercase.	title="The French Revolution " title.istitle()

isupper()	Returns True if all the character are uppercase.	<code>print("HELP".isupper())</code>
join()	Returns a string with the elements of an iterable (list, string, tuple) joined for a specified character like empty space, comma or just joined without space.	<code>items='12345'</code> <code>joined = " _ ".join(items)</code> <code>print(joined)</code>
lower()	This method returns all the string characters in lowercase.	<code>print("HELP".lower())</code>
lstrip()	This method strips the leading specified characters or space.	<code>string="The Madonna of the Pinks"</code> <code>print(string.lstrip("The"))</code>
replace()	This string method replace a word, letter of a string with other word, letter, space.	<code>print("The Arnolfini Portrait, Jan van Eyck".replace(",","-"))</code> <code>print("The Arnolfini Portrait,Jan,van Eyck".replace(","," "))</code>
rsplit()	The <code>rsplit()</code> method returns a list of strings after breaking the given string starting from the right side and understanding the blank spaces as separator if we don't specified the separator.	<code>pages="pages 396-97"</code> <code>print(pages.rsplit())</code> <code>pages="pages,396-97"</code> <code>print(pages.rsplit(","))</code>
rstrip()	The <code>rstrip()</code> method returns a copy of the string with trailing characters removed, the characters must be specified. If no argument is passed, it removes trailing spaces.	<code>pages="pages 396-97"</code> <code>print(pages.rstrip("97"))</code>
split()	<code>split()</code> method in Python separates a string into a list of strings after breaking the given string by the specified separator, if we don't do this it separates by blank space.	<code>string="The Madonna of the Pinks"</code> <code>print(string.split())</code> <code>string="The-Madonna-of-the-Pinks"</code> <code>print(string.split("-"))</code>
splitlines()	The String <code>splitlines()</code> method is used to split the lines at line boundaries. It returns a list of strings, including the line break.	<code>text="The city of Troy was founded on the western tip of Anatolia in the 19th century,and while it is almost certainly the Homeric Troy"</code>
startswith()	The <code>startswith()</code> method returns True or False, True if the string starts with the specified value.	<code>definition="Lithography is a planographic (flat-surface) method of printing"</code> <code>print(definition.startswith("Litho"))</code>
strip()	<code>strip()</code> is an inbuilt that returns a copy of the string with both leading and trailing(beginning and ending) characters removed if we have white spaces it removes it but also removes specified characters .	<code>strings=" The Madonna of the Pinks"</code> <code>print(string.strip())</code> <code>strings="The Madonna of the Pinks"</code> <code>print(strings.strip("The"))</code>
swapcase()	The string <code>swapcase()</code> method reverses or swap the Letter case of the string, it returns all uppercase characters converted to lowercase and vice versa.	<code>print("STONE-AGE sculpture".swapcase())</code>
title()	The <code>title()</code> function is a string Method which is used to convert the first character of each word to Uppercase.	<code>print("mask from mummy-case of Tutankhamun".title())</code>
upper()	<code>upper()</code> method converts all the characters in a string into uppercase characters and returns it.	<code>print("mask from mummy-case of Tutankhamun".upper())</code>

**3. Python list methods:
describe each method and provide an example**

11 points

Method	Description	Example
append()	The append() method is used for appending and adding elements to the end of the List. It changes the list.	years=[1920,1930,1940,1950] years.append(1960) print(years)
clear()	The clear() method delete all the elements from a list. It also changes the list.	pets=["dog","cat","parrot"] pets.clear() print(pets)
copy()	Returns a copy of the specified list. It is used to create other variable equal to other list.	painters=["Michelangelo","Da vinci", "Bernini"] sculptors=painters.copy() print(sculptors)
count()	This method counts the occurrences of a specified value of the elements of the list. Returns the number of elements of the specified value in the argument.	grades = [15, 20, 2, 9, 15, 7, 15, 3, 15, 15, 10,2] grades.count(15)
extend()	Add the elements of a list (or string, tuple) to the end of the current list.	authors=["Charles Perrault", "Dr. Seuss", "Lewis Carroll"] authors2=['Roal Dahl', 'Judy Blume', 'Astrid Lindgren'] writers_book_children=authors.extend(autho rs2) print(authors)
index()	Returns the first index of the element of a list. The value is specified in the argument.	lotery=[20,9,25,2,6,9,15,5,9] print(lotery.index(9))
insert()	Adds an element at the specified index position. The method requires two arguments the position and the element.	authors=["Charles Perrault", "Dr. Seuss", "Lewis Carroll"] authors.insert(2,"Grimm Brothers") print(authors)
pop()	Removes the element at the specified index position. The index is specified in the argument.	prices=[12.3,15.2,20,50,23,15,12.5] prices.pop(6) print(prices)
remove()	Removes the first item of the list of a specified value. The value is the argument in this method.	cities=["Florence","Padua", "Naples", "Assisi", "Rome", "Milan"] cities.remove("Naples") print(cities)
reverse()	Reverses the order of the list from the last item to the first.	cities=["Florence","Padua", "Naples", "Assisi", "Rome", "Milan"] cities.reverse() print(cities)
sort()	Sorts the items of the list. This means sort the items by descending order if the values are numeric and by alphabetic order.	salaries=[2200,1300,1500,1200,950,1560,22 00,4000,2200] salaries.sort() print(salaries)

4. Python tuple methods: describe each method and provide an example	2 points
---	-----------------

Method	Description	Example
count()	This method counts the occurrences of a specified value and returns it.	<pre>pets_runway=("bulldog", "german shepard", "chihuahua", "horse", "chow chow", "bulldog") print(pets_runway.count("bulldog"))</pre>
index()	This method searches the tuple for a specified value (given in the argument) and returns the position or index of where it was found.	<pre>renaissance=("Giotto di Bondone", "Simone Martini", "Lorenzo Monaco") print(renaissance.index("Lorenzo Monaco"))</pre>

5. Python dictionary methods: describe each method and provide an example	11 points
--	------------------

Method	Description	Example
clear()	Delete all the elements from the dictionary. This method modifies the dictionary.	<pre>grades_dict={'Didier':15,'Kevin':12,'Charlotte':18,'Amelie':16,'Sophie':13} grades_dict.clear() print(grades_dict)</pre>
copy()	This method copy all the items of a dictionary (values and keys).	<pre>childbooks={'Bill Martin':'Brown Bear', 'Dr Seuss':'Grinch', 'Roal Dahl': 'Matilda'} books_lib=childbooks.copy() print(books_lib)</pre>
fromkeys()	Returns a dictionary with specified keys and value, the value is the same for all the keys.	<pre>keys = {'Didier', 'Kevin', 'Charlotte', 'Amelie', 'Sophie'} value = ["A-12"] classroom = dict.fromkeys(keys, value) print(classroom)</pre>
get()	Returns the value of the specified key. The key is the argument of this method.	<pre>f1_championships = {'Michael Schumacher':7,'Juan Manuel Fangio':5,'Sebastian Vettel':4,'Ayrton Senna':3, 'Nico Rosberg':1} print(f1_championships.get('Juan Manuel Fangio'))</pre>
items()	Returns a list of tuples, in each tuple is the key and value pair.	<pre>movies={'Spielberg':'tiburón', 'Cameron':'Titanic', 'Capra': 'it happened one night', 'Kurosawa': 'Seven Samurai'} movies_l=movies.items() print(movies_l) for k,v in movies.items(): print(k)</pre>
keys()	This method returns a list containing the dictionary's keys.	<pre>childbooks={'Bill Martin':'Brown Bear', 'Dr Seuss':'Grinch', 'Roal Dahl': 'Matilda'} authors_chil= childbooks.keys() print(authors_chil)</pre>
pop()	Removes the value and key of a dictionary when the key is specified in the argument.	<pre>movies_by_year={1920:'Dr. Caligary', 1939:'Gone with the wind', 1950: 'Rashomon', 1952: 'Singing in the rain', 1960:'La Dolce Vita'} movies_by_year.pop(1939) print(movies_by_year)</pre>

popitem()	Removes the last key-value pair of a dictionary. It doesn't need an argument.	<pre> movies_by_year={1920:'Dr. Caligary', 1939: 'Gone with the wind', 1950: 'Rashomon', 1952: 'Singing in the rain', 1960:'La Dolce Vita'} movies_by_year.popitem() print(movies_by_year) </pre>
setdefault()	Returns the value of the specified key if the key exist but if it doesn't it returns None as value.	<pre> peru={'Capital': 'Lima', 'Currency': 'Sol', 'Official language': 'Spanish' } rivers = peru.setdefault('Rivers') capital= peru.setdefault('Capital') print('Rivers of Peru are: ',rivers) print('Capital of Peru is: ',capital) </pre>
update()	This method updates the dictionary with the specified key-value pairs of the argument. The new key-value pair are added to the last.	<pre> fl_championships = {'Michael Schumacher':7,'Juan Manuel Fangio':5,'Sebastian Vettel':4,'Ayrton Senna':3, 'Nico Rosberg':1} fl_driver={'Alain Prost':4} fl_championships.update(fl_driver) print(fl_championships) </pre>
values()	Returns a list but only with the values of a dictionary.	<pre> peru={'Capital': 'Lima', 'Currency': 'Sol', 'Official language': 'Spanish' } peru_values=peru.values() print(peru_values) for v in peru.values(): print('information of peru: ', v) </pre>

6. Python set methods: describe each method and provide an example	12 points
---	------------------

Method	Description	Example
add()	Adds an element to the set in any index order.	<pre> num_by_five = {5, 10, 15, 20, 25} num_by_five.add(30) print(num_by_five) </pre>
clear()	The clear method delete all the elements from the set. It returns an empty set because this method changes the set.	<pre> board_game={'Chess','Monopoly','Backga mmon','Go','Checkers'} board_game.clear() print('I deleted board games:', board_game) </pre>
copy()	This method returns a copy of the set.	<pre> brain_games={'Sudoku','Crossword', 'Kakuro', 'Latin Squares','Nonograms'} more_games=brain_games.copy() print(more_games) </pre>
difference()	The difference method returns a set containing the result of the difference between two or more sets, it is the equivalent to do the difference (A-B) set procedure in mathematics.	<pre> fruits_A={'banana', 'papaya','pitahaya', 'mango', 'kiwifuit'} fruits_B={'apple','papaya', 'mango','watermelon'} print('fruits_A - fruits_B: ',fruits_A.difference(fruits_B)) </pre>
intersection()	Returns a set that is the intersection of two or more sets. The method is like the mathematic procedure. The intersection is the common element of the sets.	<pre> fruits_A={'banana', 'papaya','pitahaya', 'mango', 'kiwifuit'} fruits_B={'apple','papaya', 'mango','watermelon'} print('fruits_A - fruits_B: ',fruits_A.intersection(fruits_B)) </pre>
issubset()	Returns True if a set is contained in other set. This means that all the elements of the set must be contained in the other set.	<pre> square_numbers={1,4,9} random_numbers={1,2,3,4,5,6,7,8,9,10} print('square numbers is a subset of random_numers?: </pre>

		<code>,square_numbers.issubset(random_numbers))</code>
issuperset()	Returns True if a set have all the elements of another set otherwise returns False.	<code>square_numbers={1,4,9} random_numbers={1,2,3,4,5,6,7,8,9,10} print(random_numbers.issuperset(square_numbers))</code>
pop()	Removes an element from the set without a specified order. In every execution this method will delete a different element.	<code>board_game={'Chess','Monopoly','Backgammon','Go','Checkers'} print('board_game: ',board_game) board_game.pop() print('board_game after pop: ',board_game)</code>
remove()	This method delete the specified element in the argument. It returns the set without the element.	<code>south_america = {'Peru', 'Argentina', 'Uruguay', 'Brazil', 'Mexico'} south_america.remove('Mexico') print(south_america)</code>
symmetric_difference()	This method execute the symmetric differences of two sets and returns a set with the results.	<code>vowels = {'a', 'e', 'i', 'o', 'u'} alphabet = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'} print(vowels.symmetric_difference(alphabet))</code>
union()	This method unites all the elements of the specified sets in the argument. It returns a set with all the elements of this sets.	<code>South_america = {'Peru', 'Argentina', 'Uruguay', 'Brazil', 'Colombia', 'Ecuador', 'Chile', 'Paraguay', 'Trinidad y Tobago'} North_america={'Canada', 'United States', 'Mexico'} Central_America={'Panama', 'Costa Rica', 'El Salvador', 'Nicaragua', 'Honduras'} print('The countries of America are: ', South_america.union(North_america, Central_America))</code>
update()	Update the set with another set, or any other iterable	<code>french_cinema = {'Godard': 'Breathless', 'Truffaut': 'The 400 Blows'} french_filmmakers = {'Resnais', 'Varda', 'Renoir'} french_filmmakers.update(french_cinema) print('french filmmakers =', french_filmmakers)</code>

7. Python file methods: describe each method and provide an example	5 points
--	-----------------

Method	Description	Example
read()	The read() method allow us to read a file content. This method returns all the file content.	with open('art.txt', 'r') as art_file: arts=art_file.read() print(arts)
readline()	This method is similar to read() because it allow us to read the file but it returns just one line of the file.	with open('Boticelli.txt', 'r') as Boticelli_file: boticelli_text_r=Boticelli_file.readline() print(boticelli_text_r)
readlines()	This method read the lines of the specified file content and returns all the lines of it.	with open('Boticelli.txt', 'r') as Boticelli_file: boticelli_text=Boticelli_file.readlines() print(boticelli_text)
write()	This method writes a string in a file.	art = 'The word “Renaissance” (meaning “rebirth”) was first used in the 15th century' with open('art.txt', 'w') as art_file: art_file.write(art)
writelines()	This method is similar to write() but writes a list of strings in a file not only a string.	Boticelli=["Known as Sandro, Alessandro di Mariano Filipepe Botticelli was a ", "\nmajor Florentine artist, at a key moment,", "\nwho emphasized ornament and sentiment rather than", "\nscientific observation and realism"] with open('Boticelli.txt', 'w') as Boticelli_file: Boticelli_file.writelines(Boticelli)