Algorithms & Data Structures Sample Exam Questions

Determine if a given string is a palindrome

A palindrome is sequence that reads same backwards as forwards. “Anna”, for example is a palindrome. The task is to determine if a given string is a palindrome.

10	ten possible points in total
60%	for explaining the algorithm you would like to use, either as narrative or in pseudo code. The explanation should include an estimate of time complexity of your algorithm.
35%	for implementing the algorithm in Python
5%	for complying with PEP 8 and PEP 20

Find all palindromic sections of any given string longer than one

A palindrome is a sequence that reads the same backwards as forwards. “Anna”, for example is a palindrome. The task is to find all palindromic sequences within any given string. The word “intellect” for example includes following palindromic sections:

“ll”	inte**ll**ect
“elle”	int**elle**ct

20	20 possible points in total
70%	for explaining the algorithm you would like to use, either as narrative or in pseudo code. The explanation should include an estimate of time complexity of your algorithm.
25%	for implementing the algorithm in Python
5%	for complying with PEP 8 and PEP 20

Print out all Fibonacci numbers smaller or equal to a given number

Given a positive number, print out all Fibonacci numbers smaller or equal to that number.
For example, given the number 11 the program should print out: 1 1 2 3 5 8
The next Fibonacci number would be 13 which is already larger than 11.

8	8 possible points in total
45%	for explaining the algorithm you would like to use, either as narrative or in pseudo code.
45%	for implementing the algorithm in Python

10%	for complying with PEP 8 and PEP 20

Determine if a sequence of numbers is increasing

Given a sequence of numbers, determine if each element is larger than the previous element.

8	Eight possible points in total
45%	for explaining the algorithm you would like to use, either as narrative or in pseudo code. The explanation should include an estimate of time complexity of your algorithm.
45%	for implementing the algorithm in Python
10%	for complying with PEP 8 and PEP 20

Determine the longest sequence of increasing numbers in a list of number

Given a sequence of numbers, determine the longest streak of consecutive numbers that are increasing. For example given the sequence [1, 2, 5, 3, 8, 9, 13, 24, 21], the longest sequence is [3, 8, 9, 13, 24].

15	15 possible points in total
70%	for explaining the algorithm you would like to use, either as narrative or in pseudo code. The explanation should include an estimate of time complexity of your algorithm.
25%	for implementing the algorithm in Python
5%	for complying with PEP 8 and PEP 20

Determine given two strings if one is an anagram of the other

Given two strings, determine if one is an anagram of the other. For example, the two strings “anagram” and “margana” are anagrams of each other.

10	10 possible points in total
45%	for explaining the algorithm you would like to use, either as narrative or in pseudo code. The explanation should include an estimate of time complexity of your algorithm.
45%	for implementing the algorithm in Python
10%	for complying with PEP 8 and PEP 20

Implement the Vigenere Cipher

Although the idea of the Vigenere Cipher is very basic it took humanity over 300 years to break it. Ultimately, frequency analysis broke it. Develop the class to encrypt and decrypt using the cipher developed by Vigenere.

8	8 possible points in total
50%	for explaining the code structure and algorithm you would like to use, either as narrative or in pseudo code
40%	for implementing the algorithm in Python
10%	for complying with PEP 8 and PEP 20

Calculate the frequency of items in a given sequence

Given a sequence of elements (that support equality function), calculate the frequency of each distinct element. For example, given the sequence [‘hi’, ‘I’, ‘am’, ‘Alexa’, ‘I’, ‘would’, ‘just’, ‘like’, ‘to’, ‘say’, ‘hi’]. The output should be:

hi	2
I	2
am	1
Alexa	1
would	1
just	1
like	1
to	1
say	1

Find two numbers in a sequence to add up to a given number

Given a sequence of numbers find any two numbers that add up to another given number. For example, given the sequence [3, 4, 1, 7 , 9, 17] and the number 8, the solution is [1, 7], because these two add up to 8.

15	15 possible points in total
60%	for explaining the algorithm you would like to use, either as narrative or in pseudo code. The explanation should include an estimate of time complexity of your algorithm.
35%	for implementing the algorithm in Python
5%	for complying with PEP 8 and PEP 20