Introduction to Programming (CC1024) 2024/2025

Exam Sample Questions

tudent ID:
full name:
Example 1 - Multiple Choice Questions
1.1. What does the following code snippet create?
a = {}
A. An empty list. B. An empty set. C. An empty dictionary. D. An empty tuple.
1.2. What is the result of the following expression?
4 % 3
A. 0. B. 1. C. 1.33333333. D. An error.

Example 2 - Short-Answer Questions

2.1. What is the meaning and functionality of the following code snippet?

if __name__ == "__main__":

2.2. Write down the output of the following code snippets:

a)	b)
<pre>def my_sum(a, b): return a + b</pre>	attack_points = 100 health_points = 20
<pre>print(my_sum(2, 3))</pre>	<pre>print(attack_points != 0 and</pre>

Example 3 - Concept Association Questions

3.1. Link the concepts on the left to keywords on the right:



Example 4 - Fill-in Blanks

 ${\bf 4.1.}$ Fill in the blanks to correctly extract and modify values in the list:

 $my_{list} = [10, 20, 30, 40, 50]$

a) Extract the third element:

x = my_list[

b) Extract a sublist containing the second, third, and fourth elements:



c) Replace the last element with 100:

my_list[] = 100

Example 5 - Implementation Questions

5.1. Interpret and correctly implement the following flowchart:



num1 = int(input("Enter a number: "))

5.2. Write a function square(s) that given a side of size s returns the area of such a square.

|--|

Name:

5.3. Consider the following code snippet:

```
i = 20
while (i >= 0):
    print(f"i={i}")
    i = i - 2
```

Write an equivalent code snippet that outputs the same result, using a for loop instead:

5.4. The following function remove_dup aims to remove repeated elements from a list:

```
def remove_dup(lst):
    for i in range(len(lst)):
        if lst[i] in lst[i+1:]:
            lst[i+1:].remove(lst[i])
    return lst
```

However, it is not working properly! Here's an example call for my_list:

my_list = [1, 2, 3, 4, 4, 4, 5, 6]
print(remove_dup(my_list))

>>> [1, 2, 3, 4, 4, 4, 5, 6]

Briefly explain why this program is not working as expected:

Example 6 - Modular Questions

To a computer scientist, the DNA strands are simply a sequence of characters formed by symbols of a 4-letter alphabet (A, T, C, G) and there are multiple operations we can perform over them.

6.1. Each DNA strand has a complementary strand, formed as follows:

A corresponds to T T corresponds to A C corresponds to G G corresponds to C

For example, the DNA strand "ACTGCATG" has the following complementary strand "TGACGTAC".

a) Write a function complement (dna_strand) that, given a DNA strand as a string, returns its complementary strand. Assume there are no invalid values/letters in the input DNA sequence string.

Example:

```
print (complement ("ACTGCATG"))
```

>>> TGACGTAC

6.2. Multiple alignment is also a classic problem when working with DNA strands. For instance, two sequences of variable length are aligned by introducing gaps ("-") in a way that the distance between them is minimal.

a) The distance between two DNA sequences is given by the total number of positions where they **do not match**. For instance, the distance between "AATGCT" and "A-TTC-" is 3. Write a function distance (seq1, seq2) that returns the distance between two DNA strands, seq1 and seq2.

Example:

```
print(distance("AATGCT", "A-TTC-"))
```

>>> 3

Student ID					Name:

b) From sequence alignment, a new sequence can be created, called "consensus" sequence. The consensus sequence between n sequences is composed by the most frequent element found in each position (excluding the gaps). For instance, aligning "AATGCT" with "A-TTC-" and "---TCC" would result in the consensus sequence "**AATTCC**". In the event of **ties** we can assume that the last valid element found is used.

	Position:	1	2	3	4	5	6	
	Seq 1:	А	А	Т	G	С	Т	
	Seq 2:	А	-	Т	Т	С	-	
	Seq 3:	-	-	-	Т	С	С	
Most Frequent:		А	А	Т	Т	С	С	

Write a function consensus (lst) that given a list lst of DNA strands, returns their consensus sequence. Assume that all sequences are already aligned, i.e., they may contain gaps ("-") but have the same length:

Example:

print (consensus (["AATGCT", "A-TTC-", "---TCC"])

>>> AATTCC

c) Finally, the distance of a consensus sequence to a list of sequences is given by the sum of individual distances between the consensus sequence and each of the other sequences. For instance, the distance between "AATTCC" and ["AATGCT", "A-TTC-", "---TCC"] would be 2 + 2 + 3 = 7. Write a function consensus_distance (seq, lst) that given a consensus sequence seq and a list lst of DNA sequences, returns the consensus distance.

Example:

```
print (consensus_distance ("AATTCC", ["AATGCT", "A-TTC-", "---TCC"])
```

>>> 7

(this is just a selection of sample exam questions aimed towards showing the type of questions you might encounter)

(the number and difficulty of questions of the real exam will be calibrated for its duration)

(you will be able to (pre)choose a portuguese or an english version of the exam)