**Assessment**

 Task .

The partial program below creates a list of the days of the week and assigns it to days.

|  |  |
| --- | --- |
| 123 | days = ["Monday", "Tuesday", "Wednesday",  "Thursday", "Friday", "Saturday", "Sunday"] |

**Question 1:** If the lines below were added to the program, what would the program output be?

|  |  |
| --- | --- |
| 45 | item = days[3]print(item) |

|  |  |
| --- | --- |
| **Answer** |  |

**Question 2:** If the lines below were added to the program, what would the program output be?

|  |  |
| --- | --- |
| 456 | index = 1item = days[index+1]print(index, item) |

|  |  |
| --- | --- |
| **Answer** |  |

**Question 3: Fill in the gaps** in the code below, so that the items in the list of days are displayed on the screen:

|  |  |
| --- | --- |
| 45 | for in : print( ) |

 Task .

Read the program below:

|  |  |
| --- | --- |
| 123456789 | planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune"]index = int(input())planet = planets[index]if index < 3:  print("It’s a rocky planet")elif planet[1] in [‘a’, ‘e’]: print("The second letter is", planet[1]) |

**Question 1:** If the user types 1 on the keyboard, what will be the output of this program when it is executed?

|  |  |
| --- | --- |
| **Answer** |  |

**Question 2:** If the user types 3 on the keyboard, what will be the output of this program when it is executed?

|  |  |
| --- | --- |
| **Answer** |  |

**Question 3:** Suppose that the elif in line 8 is **modified** to an if. If the user types 1 on the keyboard, what will be the output of this program when it is executed?

|  |  |
| --- | --- |
| **Answer** |  |

**Question 4:** The ‘rocky planets’ are the first four planets of the solar system. How would you modify the program so that all planets are correctly classified (i.e. the message “It’s a rocky planet” is displayed for all four of them).

|  |  |
| --- | --- |
| **Answer** |  |

 Task .

Read the program below and answer the questions. You can use the reference table at the bottom of the page to see how the index method works.

|  |  |
| --- | --- |
| 1234 | alphabet = 'abcdefghijklmnopqrstuvwxyz'letter = 'c'position = alphabet.index(letter)print(position) |

**Question:** What will be the output of this program?

|  |  |  |  |
| --- | --- | --- | --- |
| * 1. 2
	2. 3
	3. d
	4. ‘c’ + 1
 |

|  |  |
| --- | --- |
| **Answer** |  |

 |

**Question:** The program below is an **extension** of the previous one. **Fill in the gaps** so that the program prompts the user to enter a word and then iterates over each letter in the word to display its position in the alphabet.

|  |  |
| --- | --- |
| 123456 | alphabet = 'abcdefghijklmnopqrstuvwxyz'print("Enter a word:") = input() for in : position = alphabet.index(letter) print(position) |

**Reference table**

|  |  |  |
| --- | --- | --- |
|  | string.index(substring)e.g. word.index("intro")e.g. name.index("A") | Search in a string for the first occurrence of a substring and return its (zero-based) index.  |

 Task .

The program below converts a binary number to its decimal equivalent. The questions that follow **do not require you to understand how it works**.

|  |  |
| --- | --- |
| 12345678 | binary = input()decimal = 0weight = 1for digit in reversed(binary): if digit == '1': decimal = decimal + weight weight = 2 \* weightprint(binary, "=", decimal) |

**Question 1:** Locate all the variables in the program.

|  |  |
| --- | --- |
| **Variables** |  |

**Question 2:** Locate the line of code where program execution is suspended until the user enters a value on the keyboard.

|  |  |
| --- | --- |
| **Line** |  |

**Question 3:** Locate the line of code that contains a condition (a logical expression).

|  |  |
| --- | --- |
| **Line** |  |

**Question 4:** Locate the line of code that doubles the value of a variable.

|  |  |
| --- | --- |
| **Line** |  |

**Question 5:** Locate the lines of code that may be executed more than once.

|  |  |
| --- | --- |
| **Lines** |  |

**Question 6:** Locate the line of code that may never be executed.

|  |  |
| --- | --- |
| **Line** |  |

 Task .

Read the program below and answer the questions. The reference table after the questions shows you how the append and pop methods work.

|  |  |
| --- | --- |
| 1234567 | days = []while len(days) < 12: days.append(31) days.append(30)days[1] = 28days.pop()days.insert(7, 31) |

**Question 1:** What will be the contents of the days list after lines 1 to 4 of the program have been executed?

|  |  |
| --- | --- |
| **Answer** |  |

**Question 2:** What will be the contents of the days list after line 5 of the program has been executed?

Write the items of the list, or describe how the list will change with respect to your previous answer.

|  |  |
| --- | --- |
| **Answer** |  |

**Question 3:** What will be the contents of the days list after line 6 of the program has been executed?

Write the items of the list, or describe how the list will change with respect to your previous answer.

|  |  |
| --- | --- |
| **Answer** |  |

**Question 4:** What will be the contents of the days list after line 7 of the program has been executed?

Write the items of the list, or describe how the list will change with respect to your previous answer.

|  |  |
| --- | --- |
| **Answer** |  |

**Reference table**

|  |  |  |
| --- | --- | --- |
|  | list.append(item)e.g. numbers.append(42) | Add an item to the end of the list |
|  | list.insert(index, item)e.g. cities.insert(2, "Oslo") | Insert an item at a given position |

Resources are updated regularly — the latest version is available at: [ncce.io/tcc](http://ncce.io/tcc).

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [ncce.io/ogl](http://ncce.io/ogl).