Knowledge Check 2

Instructions: Answer the questions below, saving often. Submit your Word document to MyLO for initial feedback, then discuss your answers with your tutor in class.

Name: Sachin Kharel

Student ID: 689206

Question 1

Create a tracing table showing the execution of the following Python statements.

```
1     prism = 5
2
3     print(f"{prism} is:")
4     if prism > 7:
5       print(f"greater than 7")
6     else:
7       print(f"less than or equal to 7")
8     print("and")
9     if prism > 8:
10      print(f"greater than 8")
11    else:
12      print(f"less than or equal to 8")
```

Answer:

Line	prism	output				
1	5					
3		5 is:				
7		less than or equal to 7				
8		and				
12		less than or equal to 8				

Question 2

Create a tracing table showing the execution of the following Python statements.

```
1     prism = 8
2
3     print(f"{prism} is:")
4     if prism > 8:
5       print("larger than 8")
6     else:
7       print("not larger than 8")
8       print("and")
9       if prism < 12:
10        print("smaller than 12")
11      else:
12        print("not smaller than 12")
```

Answer:

Line	prism	output				
1	8					
3		8 is:				
7		not larger than 8				
8		and				
10		smaller than 12				

Question 3

The periodic table provides an arrangement of known chemical elements, organised by increasing atomic number. Each element in the table is represented by its atomic number,

abbreviation, full name, and weight. The following function provides summary details of an element in (a subset of) the table matching a given abbreviation.

Create a tracing table showing the execution of the function when it is called with '**Al**'. Start the tracing table at line 1 showing that `abbreviation` is assigned this value.

```
1   def element_details(abbreviation: str):
2     """Displays information for known chemical elements."""
3     element: str
4     number: int = -1
5
6     abbreviation = abbreviation.lower()
7     match abbreviation:
8       case "h":
9         number = 1
10        element = "Hydrogen"
11      case "he":
12        number = 2
13        element = "Helium"
14      case "li":
15        number = 3
16        element = "Lithium"
17      case "be":
18        number = 4
19        element = "Beryllium"
20      case "b":
21        number = 5
22        element = "Boron"
23      case "c":
24        number = 6
25        element = "Carbon"
26      case "n":
27        number = 7
28        element = "Nitrogen"
29      case "o":
30        number = 8
31        element = "Oxygen"
32      case "f":
33        number = 9
34        element = "Fluorine"
35      case "ne":
36        number = 10
37        element = "Neon"
38      case _:
39        element = "not currently supported"
40
41    if number == -1:
42      print(f"{abbreviation} is {element}.")
```

```
43        else:
44            print(f"{abbreviation} is {element} with an atomic number of {number}.")
```

Answer:

Line	abbreviation	number	element	output
1	AI			
4		-1		
6	ai			
39			not currently supported	
42				ai is not currently supported

Question 4

Create a tracing table showing the execution of the following Python statements.

```
1     sphere = 10
2
3     print(f"Before while loop sphere is {sphere}")
4     while sphere < 17:
5         print("Inside the loop")
6         if sphere >= 13:
7             sphere += 1
8         else:
9             sphere += 2
10    print(f"After while loop sphere is {sphere}")
```

Answer:

Line	sphere	output				
1	10					

3		Before while loop sphere is 10				
5		Inside the loop				
9	12					
5		Inside the loop				
9	14					
5		Inside the loop				
7	15					
5		Inside the loop				
7	16					
5		Inside the loop				
7	17					
10		After while loop sphere is 17				

Question 5

Create a tracing table showing the execution of the following Python statements.

```
1     record = {"a": "Archery", "b": "Baseball", "d": "Dodgeball"}
2
3     print("That doesn't seem quite right.")
4     record["d"] = "Diving"
5     record["a"] = "Athletics"
6     print("Need one more.")
7     record["c"] = "Cricket"
8     print("Much better. Contents of record:")
9     for k in record:
10      print(f"{k} : {record[k]}")
```

Answer:

Line	record(“a”)	record(“b”)	record(“d”)	output	record(“c”)	
1	Archery	Baseball	Dodgeball			
3				That doesn't seem quite right.		
4			Diving			
5	Athletics					
6				Need one more.		
7					Cricket	
8				Much better. Contents of record:		
10				a : Athletics		
10				b : Baseball		
10				d : Diving		
10				c : Cricket		

Question 6

Create a tracing table showing the execution of the following Python statements. Some of the variable names are intentionally vague.

```
1     source: str = "kukuninka"
2     q: str = "K"
3     n: int = 0
4     p: int = 0
5
6     source = source.upper()
7     while p < len(source):
8       if source[p] == q:
9         n += 1
10      p += 1
11
12    if n == 0:
13      print(f"{source.lower()} does not contain any {q}'s.")
14    else:
15      print(f"Found {n} occurence(s) of {q}.")
```

Answer:

Line	source	q	n	p	output	
1	"kukuninka"					
2		"K"				
3			0			
4				0		
6	“KUKUNINKA”					
9			1			
10				1		
10				2		
9			2			
10				3		
10				4		
10				5		
10				6		
10				7		

9			3			
10				8		
10				9		
15					Found 3 occurence (s) of K.	

Question 7

Create a tracing table showing the execution of the following Python statements.

```
1    mystery = [ 6, 8, 9, 2, 0, 6 ]
2    adj = 3
3
4    for a, b in enumerate(mystery):
5      if b % 2 == 0:
6        mystery[a] = b - adj
7      else:
8        mystery[a] = b + adj
```

Answer:

Line	mystery	adj	mystery [0]	mystery [1]	mystery [2]	mystery [3]	mystery [4]	mystery [5]
1	[6, 8, 9, 2, 0, 6]							
2		3						
6			3					
6				5				
8					12			
6						-1		
6							-3	
6								3

Question 8

Consider the following incomplete Python function, which you need to **complete by writing the missing 3 lines of code** so that it:

1. prompts for and reads the user's response for the loan amount into the most appropriate variable; and
2. prompts for and reads the user's response for the interest rate into the most appropriate variable; and
3. prints a message that makes use of the two variables in the designated output format.

Adhere to the following output format: if the user enters the values **550000** and **6.24** then the output should be:

```
Your recent home loan application for $550000 was approved at a
rate of 6.2% pa.
```

Take note of the number of decimal places in the expected output format.

```
1  def some_function():
2      amount: int
3      rate: float
4      amount = int(input("Enter the loan amount"))
5      rate = float(input("Enter the interest rate"))
6      print(f"Your recent home loan application for ${amount} was approved at a rate of
   {rate}% pa.")
```

Extra line numbers have been provided as answer space.

Question 9

Complete the implementation of the `converter` function below and then briefly explain why you chose the approach you did.

The function's purpose is to translate TV channel numbers into their descriptive channel name, where the channel number is provided as a parameter to the function. Use the following table as a reference for how the function should behave:

Channel number	Channel name

2, 20	ABC
6, 60, 61	Seven
8, 88	Nine
5, 50, 51	Ten
34, 36	NITV

If the channel number is not listed above, the function should return "Unknown channel".

```
1   def converter(channel: int) -> str:
2     """TODO: This function will return the channel name according to the channel number
3   input"""
4     result: str
5   if channel == 2 || channel == 20:
6     result = “ABC“
7   elif channel == 6 || channel == 60 || channel == 61:
8     result = “Seven“
9   elif channel == 8 || channel == 88:
10    result = “Nine“
11  elif channel == 5 || channel == 50 || channel == 51:
12    result = “Ten“
13  elif channel == 34 || channel == 36:
14    result = “NITV”
15  else:
16    result = “Unknown channel”
17
18   return result
```

Ensure you have changed the function documentation comment from its `TODO:` placeholder.

Extra line numbers have been provided as answer space (feel free to insert more if needed).

Finally, **explain in 1–3 short sentences** why you implemented the function that way.

Answer:

As there were only 5 cases and 1 else case, if it was easier to use {if, elif, else} approach as we were aware of what channel number has respective channel name and the text to display in else case, if it were to be more or foreseeable additions, I would use dictionary.