Grade 9 Computing

Unit 3 · Python: Lists, Strings and Dictionaries

ANSWER KEY · FOR TEACHERS

Full marks 38

Give the marks for the points in the marking scheme; an answer in other words that says the same is right.

1 Key words

1. list
2. index
3. slice
4. immutable
5. dictionary
6. key
7. KeyError
8. tuple

Each carries 0.5 mark. K · 4

2 Label the diagram

1. prices[1] → 60
2. keys
3. menu[”coffee”] → 60
4. values

Each carries 0.5 mark. U · 2

3 Multiple choice

i. **b.** `"okh"` U · 1

P is at 0, o at 1, k at 2, h at 3, a at 4. The slice takes positions 1, 2 and 3, and stops before 4: `"okh"`.

ii. **c.** `['a', 'b', 'c']` U · 1

With nothing in its brackets, `split()` cuts at every run of spaces, however long, so the two spaces count as one cut. (`"a b  c".split(" ")`, cutting at each single space, would give `['a', 'b', '', 'c']`.)

iii. **b.** 0 A · 1

`"rulers"` isn’t a key, so `get()` gives back the default given to it, 0. Without a default it would give `None`; only `stock["rulers"]` stops with a KeyError.

iv. **b.** `open("diary.txt", "a")` U · 1

`"a"` (append) writes after what is already in the file. `"w"` would wipe the earlier days first, `"r"` only reads, and `"x"` stops with an error because the file is already there.

v. **b.** a dictionary, key the district, value its headquarters A · 1

Each look-up is by a name, so a dictionary fits: `hq["Kaski"]` finds it in one step. Two lists would have to be searched and kept in step.

4 What does it print?

a. MARKING SCHEME U · 2

- ✓ `[4, 5, 1]`: 9 appended, 5 inserted at position 1, the last item popped, then 7 removed.
- ✓ 9, the item that `pop()` took off the end.

b. MARKING SCHEME U · 2

- ✓ `Bhak pur` (the first four characters, then the last three).
- ✓ 2 (`"BHAKTAPUR"` has two A’s; the small a’s became capitals).

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking

c. MARKING SCHEME A · 3

- ✓ `tea 30`: line 3 changed tea's price (the key comes once).
- ✓ `coffee 60` on the next line, and no line for momo (150 is not under 100).
- ✓ `3 True False`: three pairs; `in` looks at the keys, and 150 is a value, not a key.

5 Trace the program

ANSWER A · 4

r	total	wet	Output
(before)	0	[]	
0	0	[]	
12	12	[12]	
30	42	[12, 30]	
5	47	[12, 30]	
			47 [12, 30]

MARKING SCHEME

- ✓ `total` 0, 12, 42, 47: every round adds `r`.
- ✓ `wet` gains 12, then 30.
- ✓ `wet` doesn't change in the rounds for 0 and 5 (not over 10).
- ✓ Output `47 [12, 30]`, once, after the loop.

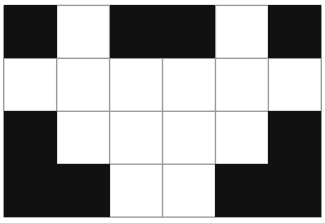
6 Find and fix the bug

MARKING SCHEME A · 3

- ✓ Line 4, a KeyError (for `'the'`).
- ✓ Reading a key that isn't in the dictionary yet: a new word has no count to add 1 to.
- ✓ `counts[w] = counts.get(w, 0) + 1`, or an if/else that sets a new word's count to 1 and adds 1 to a word already there.

7 A photo as numbers

ANSWER A · 3



new is the photo upside down (row r comes from row 3 − r) and as a negative (`255 -` swaps black and white).

```
OUTPUT
[0, 255, 0, 0, 255, 0]
```

MARKING SCHEME

- ✓ The rows in reverse order: new row 0 comes from the photo's last row, row 3.
- ✓ Black and white swapped: each pixel black where the photo's was white.
- ✓ The output `[0, 255, 0, 0, 255, 0]`: new row 0, the photo's row 3 with black and white swapped.

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking

8 Write a program

ANSWER A · 4

```
sentence = input("A sentence: ")
words = sentence.split()
print(len(words), "words")
longest = ""
for w in words:
    if len(w) > len(longest):
        longest = w
print("Longest:", longest)
print(" ".join(words[::-1]))
```

OUTPUT

```
A sentence: we climbed to the top of Nagarkot
7 words
Longest: Nagarkot
Nagarkot of top the to climbed we
```

Other right ways: `max(words, key=len)` for the longest word, or `words.reverse()` before joining.

MARKING SCHEME

- ✓ The sentence read with `input()` and cut into a list of words with `split()`.
- ✓ The number of words printed with `len()`.
- ✓ The longest word found (a highest-so-far loop on `len(w)`, or `max(words, key=len)`).
- ✓ The words in reverse order (`[::-1]` or `reverse()`), joined with spaces (or printed one after another).

9 Short answers

a. **MARKING SCHEME** U · 2

- ✓ i. A list: the order matters, and the songs are gone through one after another.
- ✓ ii. A dictionary, key the district, value its headquarters: a value found by a name in one step.

b. **MARKING SCHEME** U · 2

- ✓ `b = a` doesn't copy the list: it gives the same list a second name, so a change through `b` shows through `a`.
- ✓ `b = a.copy()` (or `list(a)`, or `a[:]`) makes a separate copy.

10 Think

ANSWER HA · 2

With two lists, a name and its marks belong together only because they sit at the same position. Sorting one list, or appending to one and forgetting the other, silently gives students the wrong marks. In a dictionary, or one list of rows, each student's data stays together; a dictionary also finds a student's marks by the name in one step, with no search for the position first.

MARKING SCHEME

- ✓ Two lists match only by position, so sorting or changing one list without the other matches names with the wrong marks; one structure keeps each student's data together.
- ✓ A dictionary finds the marks by the name in one step, with no search; two lists need a search through **names** for the position first.

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking

11 Activity: A word counter

(not marked)

WHAT STUDENTS SHOULD FIND

In almost any English paragraph the most common word is the, then small words such as a, of, and and is. In Nepali the most common are short words such as र, छ, हो and पनि. The paragraph's subject shows in the words that come next. Students who strip punctuation (including the danda, ।) and use `lower()` get the right counts; without that, "The" and "the." are counted as different words.

Marks by level	Knowledge	Understanding	Application	Higher ability	Total
Marks	4	13	19	2	**38**

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking