Grade 10 Computing

# Unit 7 · How Large Language Models Work

**ANSWER KEY · FOR TEACHERS**

Full marks 30

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. tokenizer
2. vocabulary
3. context window
4. temperature
5. parameters
6. pretraining
7. fine-tuning
8. hallucination

Each carries 0.5 mark. **K · 4**

## 2 Label the diagram

1. Tokens
2. Token numbers
3. The model
4. A chance for each

Each carries 0.5 mark. **K · 2**

## 3 Multiple choice

**i.** **b.** The vocabulary would be enormous, and any word not in it couldn't be read at all. **U · 1**

Every form of every word, every name and every language would need its own token, and a new word would be unreadable. Pieces of words keep the vocabulary to a manageable size while any text, even a new name, can still be cut into known pieces.

**ii.** **c.** It picks each token by chance, weighted by the chances it worked out, so the choices can differ each time. **U · 1**

With a temperature above 0, the model rolls a weighted dice at every token. One different token early on sends the rest of the answer a different way. The model itself hasn't changed: it doesn't learn during your conversation.

**iii.** **b.** carry on any text in the style of what it was given **U · 1**

Pretraining teaches only one thing directly: predicting the next token of ordinary text. Answering helpfully and refusing harm come from fine-tuning and feedback. No language model looks facts up in a database unless a tool is added, and none remembers past conversations by itself.

**iv.** **c.** The fact was rare or missing in its training text, and it wrote a likely-looking year anyway. **A · 1**

The model learned that a year belongs in that sentence, but a fact seen once or never can't be learned. Trained to give an answer rather than "I don't know", it wrote a likely-looking one. Temperature 0 doesn't make an answer wrong; it only stops the randomness.

**v.** **b.** Its accuracy is 90% about the world but only 50% about Nepal, so it is biased against questions about Nepal. **A · 1**

70 of 100 is right all together, but that number hides the gap: 45 of 50 is 90% and 25 of 50 is 50%. A student asking about Nepal meets the 50%. (100 questions is a small test, but enough to show a gap this large is worth checking further.)



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

## 4 What does it print?

**ANSWER** U · 4

```
OUTPUT
['tall', 'kind', 'kind']
['is', 'is']
6
```

ii. 2 in 3: "kind" is in the list twice and "tall" once, so the counts are the chances.

**MARKING SCHEME**

✓ `['tall', 'kind', 'kind']`: every word that came after "is", in order, kind twice.
✓ `['is', 'is']`: "Ram" came twice, each time before "is".
✓ 6: the different words with something after them (Ram, is, tall, ., Sita, kind).
✓ ii. 2 in 3 (about 67%), because "kind" is in the list twice.

## 5 Trace the program

**ANSWER** A · 3

| share | left | i | shares |
|---|---|---|---|
| 4 | 3 | (before) | [4, 4, 4, 4, 4] |
| 4 | 3 | 0 | [5, 4, 4, 4, 4] |
| 4 | 3 | 1 | [5, 5, 4, 4, 4] |
| 4 | 3 | 2 | [5, 5, 5, 4, 4] |

```
OUTPUT
[5, 5, 5, 4, 4]
```

**MARKING SCHEME**

✓ `share` 4 (23 // 5) and `left` 3 (23 % 5), with `shares` starting as five 4s.
✓ The loop runs for `i` 0, 1 and 2, adding 1 to the first three shares.
✓ Prints `[5, 5, 5, 4, 4]`, which adds up to exactly 23.

## 6 Check the code against its requirement

**ANSWER** A · 4

```
OUTPUT
50 True
0 False
100 False
-5 False
101 False
72.5 True
```

ii. Three tests show bugs: 0 and 100 should be True (both are included), and 72.5 should be False (it isn't a whole number). Line 2 becomes `return type(m) == int and 0 <= m <= 100`, and then the program prints True for 50, 0 and 100 and False for the rest.

**MARKING SCHEME**

✓ The output: True for 50 and for 72.5; False for 0, 100, −5 and 101.
✓ 0 and 100 are wrongly False: the requirement says both are included.
✓ 72.5 is wrongly True: it isn't a whole number.
✓ A line that meets the requirement, such as `return type(m) == int and 0 <= m <= 100`.


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

## 7 Guess, or say “I don’t know”?

**ANSWER** **A · 3**

Guessing on 48 questions gets 48 ÷ 4 = 12 right and 36 wrong.

i. A wrong answer scores 0: guessing 52 + 12 = **64**; honest **52**.

ii. A wrong answer loses a mark: guessing 64 − 36 = **28**; honest **52**.

When wrong answers cost nothing, guessing always scores more, so a model trained and judged that way learns to guess, and makes up answers.

**MARKING SCHEME**

- ✓ 12 lucky guesses and 36 wrong ones.
- ✓ i. 64 guessing, 52 honest; ii. 28 guessing, 52 honest.
- ✓ Marking that gives 0 for a wrong answer teaches a model to guess.

## 8 Short answer

**MARKING SCHEME** **A · 3**

- ✓ i. Little: a wrong fact in a game for fun, easily corrected, and nobody acts on it.
- ✓ ii. Serious: the farmer may act on it at once; too much pesticide can poison people, food and streams.
- ✓ iii. Some: the student may lose marks and readers are misled, but opening the link finds it, and nobody is hurt.

## 9 Think

**ANSWER** **HA · 2**

**Pointless:** as far as anyone can tell it has no feelings to hurt, and the extra words cost tokens and electricity. **Worth it:** how we speak becomes a habit, and a habit of rudeness to something that talks like a person can spill over to people; children watching adults learn from it; and since we can’t be sure about machines in the future, some prefer to be careful. Most of the reasons are about **us**, not the machine.

**MARKING SCHEME**

- ✓ One sensible reason on each side (no feelings to hurt, or tokens and electricity; habits, children watching, or not knowing for sure).
- ✓ Says whom it matters for, with a reason: mostly the people, their habits and those around them.

## 10 Activity: Make a tokenizer

(not marked)

**WHAT STUDENTS SHOULD FIND**

Before any merge each letter is a token: the English line has 27 and the Nepali line 25. Learning from English only, the first pairs joined are h + e, t + he and a + n (from “the” and “and”), and the English line falls to about 8 tokens, almost one a word; the Nepali line stays at 25, because the tokenizer never saw Nepali. Learning from both texts, the Nepali line falls too, to 6 tokens. A tokenizer learns the pieces that were common in its own text.

| **Marks by level** | Knowledge | Understanding | Application | Higher ability | **Total** |
|---|---|---|---|---|---|
| **Marks** | 6 | 7 | 15 | 2 | **30** |



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