Unit 9 · Automate Your Work

ANSWER KEY · FOR TEACHERS

Full marks 32

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. automation
2. script
3. dry run
4. CSV
5. byte order mark
6. regular expression
7. raw string
8. robots.txt

K · 4

Each carries 0.5 mark.

2 The pieces of a pattern

1. `\d` → **c.** any digit
2. `\s` → **h.** a space, tab or new line
3. `+` → **e.** the thing before it, once or more
4. `?` → **g.** the thing before it, once or not at all
5. `[^,]` → **a.** any one character that is not a comma
6. `\b` → **d.** the edge of a word

K · 3

Each carries 0.5 mark.

3 Multiple choice

i. **b.** `Rs [\d,]+` — A · 1

`Rs [\d,]+` needs "Rs " and then digits and commas. `R\w+` also matches "Room"; `Rs \d+` stops at the comma (it finds only "Rs 2"); and `[\d,]+` alone matches neither amount whole (it leaves out the Rs), though it finds the 75 in "Room 75".

ii. **b.** `['9', '९']` — U · 1

In Python, `\d` matches any Unicode digit, Nepali digits included. Use `[0-9]` to match only 0 to 9.

iii. **b.** 661-2345 — A · 1

Three digits, a dash, four digits, and nothing else: only 661-2345. The first has seven digits after the dash; the third has no dash; and with **fullmatch**, "Tel " in front means the whole text isn't a match (re.search would find the number inside it).

iv. **b.** The site asks programs not to visit addresses starting with /results, and a polite program doesn't. — U · 1

robots.txt is a request to programs, not a lock: people with browsers can still open those pages, and a rude program could too. Following it is what a responsible program does.

v. **b.** Open the file with `encoding="utf-8-sig"`. — A · 1

Excel's CSV UTF-8 starts with a byte order mark, which plain UTF-8 reads as an invisible character, so the column's real name is `"name"`. `utf-8-sig` drops it.

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

4 What does it print?

ANSWER **A · 3**

```
OUTPUT
['12', '7', '45', '35', '9841000011']
['Rs 35']
Room ###
None
```

MARKING SCHEME

- ✓ Every run of digits, in order, as a list of strings (7:45 gives two).
- ✓ `['Rs 35']`, and `Room ###`: every digit replaced, one # each.
- ✓ `None`: the number starts 95, which the pattern doesn't allow.

5 Short answer

MARKING SCHEME **U · 2**

- ✓ The certificate's HTML and CSS are full of `{ }`, which `str.format()` would take for gaps; string.Template's gaps are `$name`, which don't clash.
- ✓ &, < and > mean something in HTML, so a value with them could show wrongly or break the page; `html.escape()` makes the browser show them as characters.

6 Trace the program

ANSWER **A · 3**

`text`	`digits` **after line 2**	`digits` **after line 4**	**Output**
+977-9801000044	9779801000044	9801000044	+977-9801000044 9801000044
98-51000055	9851000055	9851000055	98-51000055 9851000055
984100006	984100006	984100006	984100006 None

MARKING SCHEME

- ✓ Only the digits kept: 9779801000044, 9851000055, 984100006.
- ✓ The 977 taken off only the first, which has 13 digits.
- ✓ The output, with `None` for the nine-digit number.

7 Find and fix the bug

ANSWER **A · 3**

i. Only `2083-07-08.pdf` and `2083-07-09.pdf`. Both bills were renamed to `2083-07-08.pdf`; the second rename silently replaced the first, so the fee receipt is lost (the file left is the lab bill).

ii. After line 6:

```
    if new.exists():
        print("skip:", item.name, "(" + new.name + " is taken)")
        continue
```

It then prints `skip: lab bill 2083-07-08.pdf (2083-07-08.pdf is taken)`, and all three files are kept.

iii. A dry run (print the plan, rename nothing) and a test on a copy of the folder.

MARKING SCHEME

- ✓ The fee receipt is lost: both bills get the name 2083-07-08.pdf, and the second rename replaces the first without an error.
- ✓ A check with `new.exists()` before the rename, skipping (or choosing another name) when the name is taken.
- ✓ A dry run, or running it on a copy of the folder first.

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

8 Is it worth a script?

ANSWER **A · 3**

i. Each week saves 30 − 4 = 26 minutes. 3 hours = 180 minutes; 180 ÷ 26 ≈ 6.9, so in the **7th week** it has paid for itself.

ii. By hand: 40 × 30 = 1,200 minutes (20 hours). With the script: 180 + 40 × 4 = 340 minutes (5 hours 40 minutes). It saves 1,200 − 340 = **860 minutes, 14 hours 20 minutes.**

MARKING SCHEME

- ✓ 26 minutes saved a week; 180 ÷ 26 ≈ 6.9, so the 7th week.
- ✓ By hand 1,200 minutes; with the script 180 + 160 = 340 minutes.
- ✓ 860 minutes (about 14 hours 20 minutes) saved.

9 Write a program

ANSWER **A · 4**

```
import csv

with open("fees.csv", encoding="utf-8-sig", newline="") as file:
    for row in csv.DictReader(file):
        paid = int(row["paid"])
        if paid < 1500:
            print(row["name"], "owes Rs", 1500 - paid)
```

For a sheet with Aarati Shrestha 1500, Dawa Sherpa 750, Kabita Rai 1000 and Pema Lama 0, it prints:

```
OUTPUT
Dawa Sherpa owes Rs 750
Kabita Rai owes Rs 500
Pema Lama owes Rs 1500
```

MARKING SCHEME

- ✓ The file opened (with `utf-8-sig`, for a sheet saved from Excel) and read with `csv.DictReader`.
- ✓ Each row's `paid` turned into a number with `int()` before comparing.
- ✓ Only those who paid less than 1,500 chosen.
- ✓ Each printed with 1,500 − paid.

10 Think

ANSWER **HA · 2**

Every non-digit goes, the spaces and words between the two numbers too: the result is **98410000119851000022**, twenty digits, one meaningless number. It isn't 10 digits, so this program would at least refuse it; but a program that took "the first 10 digits" would happily keep the first number and lose the second. Clean one value at a time, and find the numbers first (findall), then clean each.

MARKING SCHEME

- ✓ Every non-digit goes, the words between the numbers too, so the two numbers are joined into one meaningless 20-digit number.
- ✓ Find the numbers first (with findall), then clean each one on its own.

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

11 Activity: Test a pattern

(not marked)

WHAT STUDENTS SHOULD FIND

A first pattern such as `\d{4}-\d{2}-\d{2}` matches 2083-07-08, and refuses 2083/07/08 and 83-07-08. But it also matches 2083-13-45, which is no date. Good groups notice that a test list needs texts that must **not** match, and decide whether the pattern should also check the month is 01 to 12 (`(0[1-9]|1[0-2])`) or leave that to a later check in Python, which is easier to read. A date with a one-digit month, 2083-7-8, is a choice to make, too.

Marks by level	Knowledge	Understanding	Application	Higher ability	**Total**
Marks	7	4	19	2	**32**

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