

3 – EXERCISES ON CONDITIONAL STATEMENTS

Part A — Trace by hand

Problem 1 — Volume label

What does this program print?

Listing 1: Volume label

```
1  #include <iostream>
2
3  int main() {
4      int volume = 85;
5
6      if (volume > 100) {
7          std::cout << "too loud" << std::endl;
8      } else if (volume > 60) {
9          std::cout << "loud" << std::endl;
10     } else if (volume > 30) {
11         std::cout << "medium" << std::endl;
12     } else {
13         std::cout << "quiet" << std::endl;
14     }
15
16     return 0;
17 }
```

Output:

Now answer the same question for these alternative values of `volume`:

`volume = 20` $\Rightarrow$

`volume = 60` $\Rightarrow$

`volume = 101` $\Rightarrow$

Problem 2 — Colour name

What does this program print for `red = 255`, `green = 0`, `blue = 0`?

Listing 2: Colour name

```
1  #include <iostream>
2
3  int main() {
4      int red = 255, green = 0, blue = 0;
5
6      if (red == 255 && green == 0 && blue == 0) {
7          std::cout << "pure red" << std::endl;
8      } else if (red == 0 && green == 0 && blue == 255) {
9          std::cout << "pure blue" << std::endl;
10     } else if (red == green && green == blue) {
11         std::cout << "grey" << std::endl;
12     } else {
13         std::cout << "mixed colour" << std::endl;
14     }
15
16     return 0;
17 }
```

Output:

What would be printed for each set of values below?

red=0, green=0, blue=255 ⇒

red=128, green=128, blue=128 ⇒

red=100, green=50, blue=200 ⇒

Problem 3 — What does this return?

For each call, give the value returned by `main`. Trace each branch carefully.

Listing 3: Concert ticket price

```
1  #include <iostream>
2
3  int main() {
4      int age      = 17;
5      bool has_pass = false;
6
7      int price;
8
9      if (has_pass) {
10         price = 0;
11     } else if (age < 12) {
12         price = 5;
```

```

13     } else if (age < 18) {
14         price = 10;
15     } else {
16         price = 20;
17     }
18
19     return price;
20 }

```

Return value with age=17, has_pass=false:

What would be returned for each scenario below?

age=8, has_pass=false ⇒

age=30, has_pass=true ⇒

age=18, has_pass=false ⇒

age=11, has_pass=true ⇒

Problem 4 — Nested conditionals

Trace the program and fill in all four output lines.

Listing 4: Track status

```

1  #include <iostream>
2
3  int main() {
4      bool is_playing = true;
5      bool is_muted = false;
6      int volume      = 0;
7
8      if (is_playing) {
9          if (is_muted) {
10             std::cout << "playing but muted" << std::endl;
11         } else {
12             if (volume > 0) {
13                 std::cout << "playing with sound" << std::endl;
14             } else {
15                 std::cout << "playing silently" << std::endl;
16             }
17         }
18     } else {
19         std::cout << "stopped" << std::endl;
20     }
21 }

```

```
22     return 0;
23 }
```

Output:

Now answer for each combination:

is_playing=true, is_muted=true, volume=50 ⇒

is_playing=true, is_muted=false, volume=75 ⇒

is_playing=false, is_muted=false, volume=50 ⇒

Problem 5 — Spot the bug

This program should print "warm" for orange (red=255, green=165, blue=0). It prints something else instead.

What does it actually print?

Find the bug and fix it.

Listing 5: Buggy colour temperature

```
1  #include <iostream>
2
3  int main() {
4      int red = 255, green = 165, blue = 0;
5
6      if (red > 200 && blue < 50) {
7          std::cout << "warm" << std::endl;
8      }
9      if (green > 100) {
10         std::cout << "medium" << std::endl;
11     }
12     if (blue > 150) {
13         std::cout << "cool" << std::endl;
14     }
15
16     return 0;
17 }
```

Part B — Write the code

Problem 6 — Note name (write the code)

In music, the note **A** is often tuned to a specific frequency. But here, we use a simple integer `note\_number` (1–7) to represent the seven notes of a scale: 1 = Do, 2 = Re, 3 = Mi, 4 = Fa, 5 =

Sol, 6 = La, 7 = Si.

Write a complete `main` that:

1. Declares an `int` variable `note\_number` and assigns it the value 3.
2. Uses `if / else if / else` to print the name of the note ("Do", "Re", etc.).
3. Prints "unknown" if `note\_number` is not between 1 and 7.

Expected output for `note\_number = 3`:

Problem 7 — Image brightness category (write the code)

A pixel's brightness is stored as a single integer between 0 and 255 (0 = black, 255 = white).

Write a complete `main` that:

1. Declares `int` `brightness = 210`.
2. Prints "dark" if `brightness` is below 85.
3. Prints "medium" if `brightness` is between 85 and 169 (inclusive).
4. Prints "bright" if `brightness` is 170 or above.
5. After the category, prints a second line: "valid" if `brightness` is between 0 and 255, "invalid" otherwise.

Expected output for `brightness = 210`:

What would your program print for `brightness = 50`?

And for `brightness = 300`?

Problem 8 — Band lineup (harder, write the code)

A band can perform only if:

- At least **two** members are present (integer `members\_present`), **and**
- The stage is ready (`bool` `stage\_ready`), **or** a rehearsal is happening (`bool` `is\_rehearsal`)
— a rehearsal does not need the stage to be ready.

Write a complete `main` that:

1. Declares and initialises: `members\_present = 3`, `stage\_ready = false`, `is\_rehearsal = true`.
2. Computes a `bool` `can\_perform` capturing the rules above.
3. Prints "Let's go!" if `can\_perform` is true, and "Not ready." otherwise.
4. Returns `can\_perform`.

Expected output:

Expected return value:

Test your logic: find one set of values where `can_perform` is `false` even though `stage_ready` is `true`.

Answer: