5 – Exercises on Loops

Université Jean Monnet – Saint-Étienne 2026/2027

Quick reference

Listing 1:

```
// while loop
while (condition) {
    // body runs as long as condition is true
}

// for loop
for (initialisation; condition; update) {
    // body
}
```

A `for` loop is equivalent to:

Listing 2:

```
initialisation;
while (condition) {
    // body
    update;
}
```

Part A — `while` loops

Problem 1 — Trace the loop

Fill in the table row by row, then give the final output.

Listing 3: Countdown

```
#include <iostream>

int main() {
    int beats = 4;

    while (beats > 0) {
        std::cout << beats << std::endl;
        beats = beats - 1;
    }
```

```
10
11      std::cout << "go!" << std::endl;
12      return 0;
13  }
```

Iteration	`beats` at start	printed
1	4	
2		
3		
4		
after loop		

Problem 2 — What does this return?

Compute by hand. Fill in the table, then give the return value.

Listing 4: Accumulator

```
1   int main() {
2       int track  = 1;
3       int total  = 0;
4
5       while (track <= 5) {
6           total = total + track;
7           track = track + 1;
8       }
9
10      return total;
11  }
```

`track` (start of iteration)	added to `total`	`total` after
1	1	1
2		
3		
4		
5		

Return value:

Problem 3 — How many iterations?

Without running the code, answer the questions below.

Listing 5: Volume fade

```
int main() {
    int volume = 100;

    while (volume > 10) {
        volume = volume / 2;
    }

    return volume;
}
```

Iteration	`volume` after the body
1	
2	
3	
4	

How many iterations does the loop run? **Return value:**

Reminder: integer division truncates, e.g. `11 / 2 == 5`.

Problem 4 — Spot the bug

This program should print the numbers 1 to 5, one per line. It contains a bug. Identify it and write the fix.

Listing 6: Buggy printer

```
#include <iostream>

int main() {
    int frame = 1;

    while (frame < 5) {
        std::cout << frame << std::endl;
        frame = frame + 1;
    }

```

```
11      return 0;
12  }
```

Problem 5 — Infinite loop?

Does this loop terminate? If yes, give the return value. If no, explain why it runs forever.

Listing 7: Does it stop?

```
1  int main() {
2      int note = 0;
3
4      while (note != 10) {
5          note = note + 3;
6      }
7
8      return note;
9  }
```

Terminates? **Why / return value:** .....

Problem 6 — Write the code

Write a `main` using a `while` loop that:

1. Starts with `int pixel = 0` and `int step = 30`.
2. Each iteration, prints `pixel` then increases it by `step`.
3. Stops *before* `pixel` reaches or exceeds 255.

Expected output:

Listing 8:

```
0
30
60
90
120
150
180
210
240
```

Part B — `for` loops

Problem 7 — Rewrite as `for`

Rewrite exercise 2's `while` loop as an equivalent `for` loop. The result must be identical.

Listing 9: Same accumulator, for version

```
int main() {
    int total = 0;

    // rewrite this as a for loop
    // for ( ??? ; ??? ; ??? ) { ... }

    return total;
}
```

Problem 8 — Trace the `for` loop

What does this program print?

Listing 10: Odd frames

```
#include <iostream>

int main() {
    for (int frame = 1; frame <= 10; frame = frame + 2) {
        std::cout << frame << std::endl;
    }
    return 0;
}
```

List all printed values:

How many iterations does the loop run?

Problem 9 — Nested loops

What does this program print? Write out every line of output.

Listing 11: Grid of pixels

```
#include <iostream>

int main() {
    for (int row = 0; row < 3; row = row + 1) {
        for (int col = 0; col < 3; col = col + 1) {
            std::cout << row * 3 + col << " ";
        }
        std::cout << std::endl;
    }
    return 0;
}
```

Line 1:

Line 2:

Line 3:

How many times does the inner loop body execute in total?

Problem 10 — Spot the bug

This program should print the 5 times table from 1×5 to 5×5, one result per line (5, 10, 15, 20, 25). It contains two bugs.

Listing 12: Buggy times table

```
#include <iostream>

int main() {
    for (int i = 0; i <= 5; i + 1) {
        std::cout << i * 5 << std::endl;
    }
    return 0;
}
```

Bug 1:

Bug 2:

Problem 11 — Write the code

Write a `main` using a `for` loop that:

1. Iterates over the values 10, 9, 8, . . . , 1 (counting *down*).
2. Accumulates the sum of all those values in an `int` variable.
3. Returns the sum.

Expected return value:
Hint: think carefully about the initial value, the condition, and the update step for a countdown.

Exercise 13 — Jukebox (open-ended)

A jukebox plays songs one by one. You are given two integers: `song_count` (how many songs are in the playlist) and `skip_every` (every song whose position is a multiple of `skip_every` is skipped).

Write a complete `main` that, for `song_count = 8` and `skip_every = 3`:

1. Goes through positions 1 to `song_count` inclusive.
2. Prints `"playing N"` where N is the track number
3. Prints `"skipping N"` for tracks that should be skipped, where N is the track number that is being skipped
4. After the loop, prints the total number of songs actually played (not skipped).

The first track number is 1.

Expected output:

Listing 13:

```
playing 1
playing 2
skipping 3
playing 4
playing 5
skipping 6
playing 7
playing 8
6 songs played
```

Choose whichever loop you think fits best and be ready to justify your choice. There is no single correct answer.