

Suppose all atomic variables have been initialized to 0 before the threads have been created.

1. Thread 1:

```
int a = x.load(std::memory_order_acquire);
y.store(1, std::memory_order_release);
```

Thread 2:

```
int b = y.load(std::memory_order_acquire);
x.store(1, std::memory_order_release);
```

(a) Which of the following are possible final values for the pair (a, b)?

Options: (0, 0) / (0, 1) / (1, 0) / (1, 1)

(b) What if we change the `x.load()` operation to `std::memory_order_relaxed`?

(c) What if we change both `x.load()` and `y.load()` to `std::memory_order_relaxed`?

2. Thread 1:

```
x.store(1, std::memory_order_relaxed);
y.store(1, std::memory_order_release);
x.store(2, std::memory_order_relaxed);
```

Thread 2:

```
x.store(3, std::memory_order_release);
```

Thread 3:

```
while (y.load(std::memory_order_acquire) != 1);
cout << x.load(std::memory_order_acquire);
cout << x.load(std::memory_order_acquire);
```

(a) True or False: The two `cout`-s never print 0.

(b) Which of the following are possible outputs of the two `cout`-s?

Options: 11 / 12 / 13 / 21 / 22 / 23 / 31 / 32 / 33

3. Thread 1:

```
x.store(1, std::memory_order_relaxed);
z.store(1, std::memory_order_release);
```

Thread 2:

```
z.fetch_add(1, std::memory_order_relaxed);
```

Thread 3:

```
while (z.load(std::memory_order_acquire) != 2);
cout << x.load(std::memory_order_relaxed);
```

(a) True or False: The program is guaranteed to terminate.

(b) True or False: There is a data race.

(c) True or False: The `cout` must print 1.

(d) True or False: Consider changing thread 2 to `z.load(std::memory_order_relaxed); z.store(2, std::memory_order_release)`. When the `z.load()` returns 1, the `cout` might print 0.

4. Thread 1:

```
x.store(1, std::memory_order_relaxed);
y.store(1, std::memory_order_release);
```

Thread 2:

```
x.store(2, std::memory_order_relaxed);
y.store(2, std::memory_order_release);
```

Threads 3 & 4:

```
while (y.load(std::memory_order_acquire) == 1);
cout << x.load(std::memory_order_relaxed);
```

(a) True or False: The program is guaranteed to terminate.

(b) Which of the following are possible outputs of the `cout`?

Options: 0 / 1 / 2

(c) True or False: It is possible that thread 3 prints 1 but thread 4 prints 2.

(d) True or False: If the while loop body executes at least once for both threads 3 and 4, it is possible that thread 3 prints 1 but thread 4 prints 2.

(e) For (a)-(d), what if we change the condition of the while loop to `while (... == 0)`?