Quiz 3

Questions relating to the Threading API are covered in this lesson.

Question # 1

Consider the snippet below:

```
monitor = Monitor.new

monitor.synchronize {
  monitor.synchronize {
    puts "Main thread executing"
  }
}
```

Q What is the outcome of the above program?

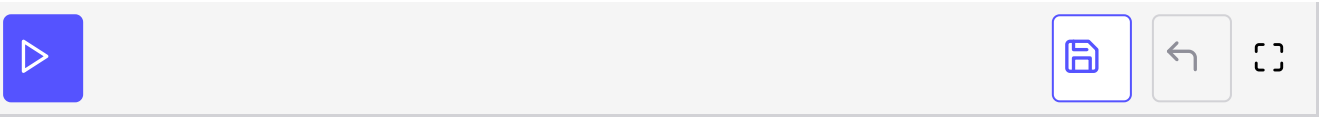
COMPLETED 0%

1 of 1

```
require 'monitor'
```

```
monitor = Monitor.new

monitor.synchronize {
  monitor.synchronize {
    puts "Main thread executing"
  }
}
```



Question # 2

Consider the snippet below:

```ruby
monitor = Monitor.new

Thread.new do
  monitor.enter()
  monitor.enter()
  puts "Child thread executing"
  monitor.exit()
end

# wait for child to execute
sleep(2)

monitor.enter()

puts "Main thread exiting"
```

Q What will be the outcome of the above program?

COMPLETED 0%

1 of 1

```
require 'monitor'

monitor = Monitor.new

Thread.new do
  monitor.enter()
  monitor.enter()
  puts "Child thread executing"
  monitor.exit()
end

# wait for child to execute
sleep(2)

monitor.enter()

puts "Main thread exiting"
```

Question # 3

The child thread from the previous snippet is slightly tweaked as follows:

```
Thread.new do
  monitor.enter()
  monitor.enter()
  puts "Child thread executing"
  monitor.exit()

  # sleep forever
  sleep()
end
```

Q What will be the outcome of running the program with the above changes?

COMPLETED 0%

1 of 1

```
require 'monitor'

monitor = Monitor.new

Thread.new do
  monitor.enter()
  monitor.enter()
  puts "Child thread executing"
  monitor.exit()

  # sleep forever
  sleep()
end

# wait for child to execute
sleep(2)

monitor.enter()

puts "Main thread exiting"
```

Question # 4

Consider the same program and now the child thread raises an exception as follows:

```
Thread.new do
  monitor.enter()

  monitor.enter()
  # raise an exception
  raise "Ka Boom"
  monitor.exit()

  # sleep forever
  sleep()
end
```

Q

What will be the outcome of the program now?

COMPLETED 0%

1 of 1

```
require 'monitor'

monitor = Monitor.new

Thread.new do
  monitor.enter()
  monitor.enter()
  raise "Ka Boom"
  monitor.exit()

  # sleep forever
  sleep()
end

# wait for child to execute
sleep(2)
```

```
puts "Main thread exiting"
```

Question # 5

Consider the snippet below:

```
monitor = Monitor.new
monitor.exit()
```

Q What will be the outcome of the above snippet?

COMPLETED 0%

1 of 1

```
require 'monitor'

monitor = Monitor.new
monitor.exit()
```