OnTrack

Coding exercise

1. Given these functions:

```
def f1 : Future[Unit] = ???
def f2 : Future[Unit] = ???
def f3 : Future[Unit] = ???
def f4 : Future[Unit] = ???
```

Write code to execute them when:

- there are no dependencies between the functions
- f4 depends on f3 which depends on f2 which depends on f1
- f4 depends on f3 and f2, and f3 and f2 both depend on f1

2. Given two functions f1 and f2, implement f3 by composing f1 and f2

```
val f1: (Int, Int) => Int = (a, b) => a + b
val f2: Int => String = _.toString
val f3: (Int, Int) => String = ???
```

3. Given a list Seq(1, 2, 3)

which represents the number 123, write a function to increment it by one without converting types. Your function should produce the expected result for the
following test cases:

```
Nil => Nil
Seq(0) => Seq(1)
Seq(1, 2, 3) => Seq(1, 2, 4)
Seq(9, 9, 9) => Seq(1, 0, 0, 0)
```

4. Given the following function:

```
def f[A](a: A): Future[A]
```

Write a function `g` that safely handles calling f. The return type of `g` should be such that when f succeeds, g returns something very similar. Feel free to import an external library for the return type of g.

5. Explain what the following code means:

```
trait MyAlg[F[_]] {
  def insertItSomewhere(someInt: Int): F[Unit]
  def doSomething(someInt: Int): F[Int]
}
```

Mention some advantages of the above code.

6. Given the trait in Q5,

create a class `MyProg` abstract in type F that has MyAlg passed to it.
Implement the following method in the class:

```
def checkThenAddIt(someInt: Int) = ???
```

It should pass the result of `doSomething` to `insertItSomewhere`.
Feel free to add external imports.

7. How would you design a REST API

for an address book? What endpoints will it have (feel free to provide sample curl requests)? How would you handle errors?