- Exercise

Let's test our knowledge of auto with this small coding exercise.

WE'LL COVER THE FOLLOWING

- Exercise 1

Exercise 1

Below, we can find the code from the previous example. We must replace the `auto` keyword with actual explicit data types.

Try to replace as many usages of `auto` as possible.

Do think of the possible headers we may need to add.

```
#include <chrono>
#include <future>
#include <map>
#include <string>
#include <tuple>

int main(){

  auto myInts = {1, 2, 3};
  auto myIntBegin = myInts.begin();

  std::map<int, std::string> myMap = {{1, std::string("one")}, {2, std::string("two")}};
  auto myMapBegin = myMap.begin();

  auto func = [](const std::string& a){ return a;};

  auto futureLambda= std::async([](const std::string& s ) {return std::string("Hello ") + s;}

  auto begin = std::chrono::system_clock::now();

  auto pa = std::make_pair(1, std::string("second"));

  auto tup = std::make_tuple(std::string("second"), 4, 1.1, true, 'a');

}
```

We can find the **solution** in the next lesson.