- Solution

In this lesson, we'll look at the solution to the exercise from the previous lesson.

WE'LL COVER THE FOLLOWING

Solution

```
#include <iostream>
#include <numeric>
#include <string>
#include <vector>

int main(){

  std::cout << std::endl;

  std::vector<int> myVec{1, 2, 3, 4, 5, 6, 7, 8, 9};
  auto res1 = std::accumulate(myVec.begin(), myVec.end(), 0);
  std::cout << "res1: " << res1 << std::endl;

  auto res2= std::accumulate(myVec.begin(), myVec.end(), 1, [](int fir, int sec){ return fir
  std::cout << "res2: " << res2 << std::endl;

  std::vector<std::string> strVec2{"Only", "for", "testing", "purpose"};

  std::string res3 = std::accumulate(strVec2.begin() + 1, strVec2.end(), strVec2[0],
                                     [](auto fir, auto sec){ return fir + ":" + sec; });

  std::cout << "res3: " << res3 << std::endl;

  std::cout << std::endl;

}
```

Explanation

- We have created a vector that we initialized with integers ranging from 1

- We have created a vector that we initialized with integers ranging from 1 to 9.
- In line 11, we have used `std::accumulate` to sum up all values from 1 to 9.
- In line 14, we have used `std::accumulate` to multiply all the elements using a lambda function.
- In lines 19 and 20, we have used `std::accumulate` to print all the elements of the vector containing strings.

That's it for the algorithms. In the next chapter, we'll learn callables in detail