EXERCISES 1.

- Write your own my_mean function that computes the mean of a vector
- It takes an integer vector of real numbers as input, and returns a real value

```
library(Rcpp)

####################################
# my_mean()
####################################

cppFunction('
double my_mean(NumericVector x) {
< WRITE YOUR CODE HERE>
Return(y);
}
')
```

test it with in R with:

```
my_mean(c(1,2,3))
[1] 2
```

```
cppFunction(
'
double my_mean(NumericVector x) {
  int n = x.size();
  double y = 0;
  int i;
  for(i = 0; i < n; ++i) {
    y = y + x[i];
  }

  y = y / n;
  return(y);
}
'
)

my_mean(c(1,2,3))
```

EXERCISES 2.

- Write your own my_cumsum function that computes accumulated sums, like:

```
my_cumsum(c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
[1]   1  3  6 10 15 21 28 36 45 55
```

```
library(Rcpp)

#####################################
# my_cumsum()
#####################################

cppFunction('
NumericVector my_cumsum(NumericVector x) {
  int n = x.size();
  NumericVector out(n);
< WRITE YOUR CODE HERE>
return(out);
}
')
```

```
##################################
# my_cumsum()
# my_cumsum(1:10)
#   1  3  6 10 15 21 28 36 45 55
##################################

cppFunction('
NumericVector my_cumsum(NumericVector x) {
  int n = x.size();
  # This creates a vector of length n
  NumericVector out(n);

  out[0] = x[0];
  int i;
  for(i = 1; i < n; ++i) {
    out[i] = out[i - 1] + x[i];
  }
  return(out);
}
')
```

EXERCISES 3.

- Write your own my_range function that computes the mínimum and máximum value of a vector, like:

my_range(c(3, 1, 5,2))

[1] 1 5

Use the sugar *max()* and *min()* R functions

```
library(Rcpp)

#####################################
# my_range()
#####################################

cppFunction('

NumericVector my_range(NumericVector x) {
  NumericVector out(2);
  <WRITE YOUR OWN CODE HERE>
  return(out);
}
')
```

```
#####################################
# my_range()
# my_range(c(3, 1, 5,2))
#   1 5
#####################################

cppFunction('

NumericVector my_range(NumericVector x) {
  # This creates a vector of length 2
  NumericVector out(2);
  out[0] = min(x);
  out[1] = max(x);
  return(out);
}
')
```