Excercise set week three

Feedback on these excercises is very much appreciated. Send mail to one of the lecturers or to Aryan at aryannm@gmail.com

Solutions are available in file `solutions.md`

Problem X: (Repetition)

1. Write a program in C++ that asks for 10 integer inputs from the user and finds both the max and min.
2. Write the above program but this time store the integer inputs in a container (a plain c int array or c++ std::vector>.

Problem One: (References)

1. Implement a function `foo` that has void return and takes as input `int& a` and multiplies this input by 10.
2. Implement a function `foo2` that has void return and takes as input `const int& a` and prints out the value of a.
3. Implement a function `foo3` that does the same thing as `foo` but with as input `const& int a`. Does this compile? Why? Why not?
4. What is the point of const references?
5. When should you use pass by value `int a` and when should you pass by reference `int& a` when creating functions?

Problem Two: References & Standard library

Implement the first program of Problem X using std::vector. Reason about which of the functions should use pass by value, pass by reference and pass by const reference.

1. Create a function that as input takes the vector you just created and prints out every other value.
2. Create a function that as input takes the vector you just created and prints out the mix and max values, use standard library functions for sorting! (use google).

Extra: Sorting

Create a function that takes as input the vector you just created and sort it by a custom sorter. Ie when you decide to sort a vector it gets sorted from

smallest to largest. But suppose you want to sort by evenness. Eg the numbers `1,22,17,3,6` will get sorted into even numbers followed by odd numbers `6,22,3,1,17`. If you research the custom sorter in C++ you will find that you can sort it however you like.

Problem Three: Alphabet frequency

Write a C++ program that count the total occurence of each letter in a given string. For example for the input "Hello World" the output should be. It is up to you how you want to implement this but the input should be std::string. I will hint at the fact that there are standard library functions that can help you count.

```
H: 1
W: 1
d: 1
e: 1
l: 3
o: 2
r: 1
```