Programming task for part 1:

Make a “hallo world” application.

Store your ages in a program using doubles

Calculate the average age.

Calculate how old each person is in month and calculate the average month per person.

Calculate the first 5 Fibonacci\* numbers and print those numbers out

A person earns 19000 kr per month, get 5 % extra in pension and pay 43% in taxes. Store these values in a program. Then write a program that calculates the following and print them out.

How much pension does the person earn per month?

How much tax does that person pay from base income, from pension and in total.

How much does he earn per year before taxes and after taxes?

How much pension does he get in a year?

The person goes on pension in 5 years and gets the entire pension at once. Calculate how much he will get when he gets 3% interest every year.

\*By definition, the first two numbers in the Fibonacci sequence are 0 and 1 and each subsequent number is the sum of the previous two.

Programming task for part 2:

Make a program that count up from 0 to 10 using either a while or a for loop

Make a program that count down from 10 to 0 using either a while or a for loop

Make a program that that calculates the value of 1+2+3+4+ … +20 using either a while or a for loop

Make a program that prints out the multiplication table for 2 to 10, using 2 nested for-loops. One for the table and one for the multiplication

Make a program that calculated the first 20 Fibonacci numbers and print them. You should either use a for or a while loop.

Use the random generator ((int)rand()) to generate 10 numbers and decide if they are even or not and print the result out. You can use modulus (eg if(random%2==0) to find out if it is even

1: Create a function that read the file dataexample.txt and print each line out to the console

2: Create a function that read the file dataexample.txt and save the first column in a vector called observation and the last column in another vector called modelOutput. (Tip: use a stringstream to separate the 2 columns)(Tip 2: reuse some code from exercise 1)

3: Move the two Vectors outside the function you create in exercise and but still keep them as part as that cpp file. Then Create a function that use a for loop to iterate trough each of the arrays and print each element out.

4: The instruments has failed and measured too high. Make a function that takes an double as argument. Subtract that number from each element in the observation Vector. And save the new data into a text file called newDataSet.txt.

5: Read the new dataset and implement this special version of RMSE\*

P is the model data and O is the observed. To use power and square root then include #include <math.h> and you can then use pow(base,power) and sqrt(item).

6: the file called “Temperature data.txt” contains daily temperature. Write some code that read those values and calculate the monthly mean and write those to a file.

7: Make sure that the code that you did write can handle missing days

8: Implement this function and use data from newDataSet.txt

P is the model data and O is the observed

\*Please note that this is a special version of RMSE and not the general one.