Exercise 1

- Create a program *ex1.c* that declares integer, unsigned short integer, signed long int, float and double variables.
- Find out how to assign maximum values for each variable (hint: use `INT_MAX` for integer, etc.)
- Print sizes and values of each variable.
- Write a script *ex1.sh* to run your program.
- **Note:** you should submit the file *ex1.c* and the script *ex1.sh*.

Exercise 2

- Write a program *ex2.c* that asks the user to type a string character-by-character until **dot (.)** character is entered and prints its reverse with double quotation.
  - Example

Input	Output
C language. is easy and simple.	"egaugnal C"

Example on Exercise 2 Input/Output

- Write a script *ex2.sh* to run your program.
- Hints:
  - a string in C is an array of chars (more about arrays next week)
  - use `strlen()` function to get the length of a string
- **Notes:**
  - You should submit the file *ex2.c* and the script *ex2.sh*.
  - The maximum size of the string is 256.
  - If the user did not type **dot (.)** character, then pressing **Enter key** will be accepted as terminating character.

Exercise 3 (1/2)

- In computer systems, the data is stored in the memory as 0s and 1s which forms a number in binary system. We use different numeral systems since reading these binary numbers is difficult for humans. The most common number systems are binary, decimal, octal, and hexadecimal systems.
- In this exercise, you have to write a function **convert** which converts a given number x from a numeral system s to another numeral system t where t, s are numbers in the range [2-10]. If the given number is wrong or s or t are out of the previous range then we should print the error message **"cannot convert!"**.

  For instance, the function call **convert(1234, 8, 2)** will convert the given number *1234* from the octal (8) system to the binary (2) system and prints *1010011100*.

Exercise 3 (2/2)

- Write a program *ex3.c* which uses the function **convert**, reads a long long number and the source and target number system specifiers from the user, then it should print the converted number or error message in case of errors.
- Write a script *ex3.sh* to run your program.
- **Notes:**
  - You should submit *ex3.sh* and *ex3.c* which contains the function **convert**.
  - We assume that the user enters a non-negative number.
  - using arrays, structures or pointers is not allowed in this exercise. But you can use an array of characters which is a string.
  - The numerals in the number systems are represented by decimal numbers [0-9]. For instance, in the number system 7, we have the numerals [0, 1, 2, 3, 4, 5, 6].
  - We did not specify the return type of the function **convert**.
- **Hint: use `sscanf()` to convert string to int and `sprintf()` to convert int to string.**

Exercise 4 (1/2)

- Write a function **count** which returns the number of occurrences of an input character in a string.
- Example:

Input	Output
Innopolis, i	i:2
Innopolis, m	m:0

Example on **count** function input/output

- Write a function **countAll** which prints the number of occurrences of each character in the input string.
  - **Hint:** Use your function **count**.
- Example:

Input	Output
Innopolis	i:2, n:2, n:2, o:2, p:1, o:2, l:1, i:2, s:1

Example on **countAll** function input/output

Exercise 4 (2/2)

- Write a program *ex4.c* that accepts an input string from the command line and prints the number of occurrences of all characters in the input string.
- Write a script *ex4.sh* to run your program.
- **Notes:**
  - using arrays, structures or pointers is not allowed in this exercise. But you can use an array of characters which is a string.
  - The maximum size of the string is 256.
  - your program should be case-insensitive. This means that characters 'i' and 'I' are treated as a single character.
  - you should submit *ex4.sh* and *ex4.c* which contains the functions **count** and **countAll**.

Exercise 5

- The Tribonacci sequence T_n is defined as follows:

$$T_n = \begin{cases} 0 & n = 0 \\ 1 & 1 \leq n \leq 2 \\ T_{n-1} + T_{n-2} + T_{n-3} & n \geq 3 \end{cases}$$

- Write a function **tribonacci** that takes as argument n and returns the value of T_n $(0 \leq n \leq 37)$
- You are neither allowed to use arrays nor function recursion
- Write a program *ex5.c* which calls the above function with arguments 4, 36 and print the output to standard output.
- Submit your ex5.c file accompanied by an ex5.sh file to compile and execute ex5.c