Electrical Engineering and Information Technology, B.Eng.
Introduction to the C Programming Language: Exercises

Exercise Sheet 12

---

1. In chapter 7.1 of the lecture we declared the struct complex for the use of complex numbers. Write a function c_swap(), which exchanges 2 complex numbers with each other. (See example E.5-4 from the lecture)

   The multiplication of 2 complex numbers z = x * y is defined by

```
z.re = x.re * y.re - x.im * y.im
z.im = x.re * y.im + x.im * y.re
```

   Write a function c_mul(), which implements the multiplication of 2 complex numbers. (See E.7-3)

   Test both functions in a main().

2. Extend exercise 1 and use typedef to define the type complex_number.

3. In E.7-5 we defined the type MasterData by

```
typedef struct {
          NameType       family_name;
          NameType       given_name;
          struct {
                    unsigned short year,
                                   month,
                                   day;
                 } birthday;
          float          salary;
        } MasterData;
```

   Write a function which reads the values for each member of a variable of type MasterData from the keyboard. Write a main() to test your function: define a variable of type MasterData, use your function to read the data for your variable and finally print the values of the members to the screen.

4. Define the type MessageTag using typedef in analogy to E.7-8 to hold the values DIGITAL_INPUT, ANALOG_INPUT and TEXT.

   Define the

```
typedef struct short_event {
          MessageTag     message_type;
          union {
               short     digital_value;
               float     analog_value;
               char      message[20];
          }payload;
} short_event;
```

a) Write a function PrintStruct() which imports a variable of type short_event and prints the message type and the correct value to the screen.
Write a main() to define a variable test_event of type short_event, fill the variable with message_type and a corresponding value. Call PrintStruct().
Printf() sizeof(test_event).
Try out all 3 types of MessageTag.

b) Write a main() to define a variable test_event of type short_event, fill message_type with DIGITAL_INPUT and set analog_value = 3.7. Call PrintStruct(): what is the result?
Printf() sizeof(test_event).

5. Try to get access to the code of E.7-11.

a) Extend the struct customer with a member `short number;`
Extend readname() to read in an int value for number.

b) Write a function with a short as a parameter, which searches in the linked list for the struct with the given number and prints the member name if a corresponding struct has been found.

c) Extend part a) of this exercise to check whether the given number is already existing in the linked list. If yes, exit with an error message.