# Embedded Aviation SW Developer Evaluation Exercises

Work on and provide solution for at least one of the following exercises.

## Exercise #1 (Data read and manipulation)

Assume a data set in a file in CSV format, with semicolons (“;”) as field separators.

Each entry is made up of fields (from left to right)

* ID (sequence of at most 32 characters)
* Name (sequence of at most 64 characters)
* Age (natural number)
* Data of at most 256 bytes, represented in hexadecimal notation (i.e. at most 512 characters)

The ID is unique in the data set, the Name not.

Solve/answer the following tasks/questions:

1. Task 1: Read the data set into memory, representing the data in an array.  
    Your implementation must not use pointers to data items other than C strings and the   
    array(s) itself.  
    Based on command input to the program perform the following actions as efficient as   
    possible:
   1. Output the data in ascending order sorted by the ID (ID compare is string compare)
   2. Output the data in descending order sorted by ID
   3. Output the data in ascending order sorted by Name
   4. Allow to search for an ID (including output of matched data)
   5. Allow to search for a Name (including output of matched data)
2. Question 1: How can the sorting be utilized to implement the search efficiently?
3. Question 2: How can re-sorting the whole data array be avoided if 1.a, 1.b and 1.c are issued repeatedly?

In your code you shall not use library routines for searching or sorting.  
Only library routines for I/O and memory management are allowed.

## Exercise #2 (Embedded Measurement Cycle)

Assume a little temperature measurement box with two inputs for PT100 temperature signals and four discrete outputs ( 2x WARN and 2x ERROR).

The function of the box shall be to constantly measure the temperature (any of the inputs will do) and activate a WARN output when the temperature raises above 40°C and a ERROR output if the temperature raises above 60°C. If temperature falls below the thresholds, discretes shall be deactivated again.

The temperature sensors are connected to two Analog Digital Converter (ADC) channels at the CPU, the discretes are realized as digital I/O pins.

Resolution of the ADC is 12 bit, the range of the temperature sensors is -50°C to 150°C.  
ADC conversion cycle takes around 0.1ms.

Answer/solve the following questions/tasks:

1. Task 1: Write down requirements and a SW design that represents the given function.
2. Question 1: What has to be considered in the chain from temperature measurement input data to the discrete output paths? What data do you need in addition to the above function description?
3. Task 2: Write a code skeleton to implement above function, using the types/functions defined in the header file below to access data/interact with the system.
4. Question 2: What influence has time on your code/design? Why?

Header file (interface.h):

#if !defined(\_INTERFACE\_H\_)

#define \_INTERFACE\_H\_

/\* enumeration for ADC channel \*/

enum e\_temp {

SENSOR\_1 = 0U,

SENSOR\_2 = 1U

};

typedef enum e\_temp temp\_t;

/\* discrete channels \*/

enum e\_disc {

CHANNEL\_WARN\_1 = 0U,

CHANNEL\_WARN\_2 = 1U,

CHANNEL\_ERROR\_1 = 2U,

CHANNEL\_ERROR\_2 = 3U

};

typedef enm e\_disc disc\_t;

/\* functions \*/

/\* get ADC value \*/

unsigned int get\_adc(temp\_t channel);

/\* set discrete active/inactive \*/

void set\_disc(disc\_t channel);

void clear\_disc(disc\_t channel);

/\* wait certain number of us (0.001ms) \*/

void delay(unsigned int us);

#endif

## Exercise #3 (DO-178C development)

Imagine the box from Exercise #2 being used on an aircraft to implement a temperature monitoring and warning function. The safety assessment of the function is DAL C for the warning and DAL B for the error signaling. Reliability of temperature measurement data is influenced by EMI effects, which may invalidate data on a single channel from time to time. Reaction time limit from measurement to signaling imposed by system requirements is 10ms. Discrete signal outputs are wired independently from each other in the system. Fault on a single line may occur with a probability of 10-5/FH.

Answer/solve the following questions/tasks:

1. Question 1: Which artefacts do you need for a DO-178C compliant development for the function? What is input to the SW development, what is created by the SW development?
2. Task 1: Write a set of high-level SW requirements covering the function according to DO-178C.
3. Question 2: What effects have the DAL levels of the system on your SW design/ requirements?
4. Task 2: Provide a SW architecture that represents your SW implementation to the function.
5. Question 3: Functions with different DALs are mixed in the box. Does partitioning make sense here to separate the functions by DAL?