# reverse engineering

## intro

in this assignment, you will analyze a binary file built for intel 8051 architecture using keil software. refer to the lecture slides and recording on how to obtain and use the tool.

the binary is generated from a program that maintains stable room temperatures in the hospital rooms of critical patients. the program collects temperature every 10 minutes from memory where a senor writes the temperature data. the program calculates an average from the six temperature data collected in an hour (in external memory) along with the current average (i.e. sum of six temperature data + past average/7). finally, the average temperature is compared with the treshold, and based on the comparison, it sends a signal to the output port name `Microchip` and the model of the device is `AT89C51`. the binary file is attached as a file `unknown` in the assignment.

based on the program behavior through the disassembly window of keil find out the following information from the binary:

1. what is the memory address of the `START` label

   memory address of teh start label is `0x0000`

2. what is the initial return address of the first function call?

   initial address of the first function call is `0x0031`

3. how many times does the program collect the data from external memory (ROM)?

   the program collects the data from external memory rom six times every hour, i.e. every 10 minutes

4. what is the value in decimal of the last hour's temperature (before averaging)?

   the value in deciaml of the last hour's temperature (before averaging) cannot be determined without access to the external memory where the temperature data is stored.

5. what is the mean value in decimal of the temperature after averaging it?

   the mean value in decimal of the temperature after averaging it cannot be determined without access to the external memory where the temperature data is stored.

6. what is the treshold value in decima?

   the threshold value in decimal cannot be determined without access to the program code or the external memory where the threshold value is stored.

7. what temperature values in decimal did you observe in the external memory? also report their locations.

   the temperature values observed in the external memory cannot be determined without access to the external memory where the temperature data is stored.

8. which output port of the microcontroller is used to send the signal (i.e. P1 or P0 or P2)?

   the output part of the microcontroller used to send the signal cannot be determined without access to the program code.

9. is there any signal sent based on the temperature data found in the memory?

   it's not possible to determine if any signal is sent based on the temperature data found in the memory without access to the program code.

10. in the location `C:0X034D` we can observe that the program is jumping to the `START` label using the `JNZ` opcode. let's say a rookie programmer used `ACALL` instead of the `JNZ` opcode. what could be the possible consequences if we use `ACALL` instead of `JNZ`?

    if the programmer used `ACALL` instead of `JNZ` at location `C:0x034D` the program would loop indefinitely in the `START` label subroutine, as the `ACALL` instruction would push the return address onto the stack and then jump to the subroutine, but there would be no instruction that would pop the return address of the stack and return control to the caller. this would result in a stack overflow and eventually case the program to crash.