Reagan Carter

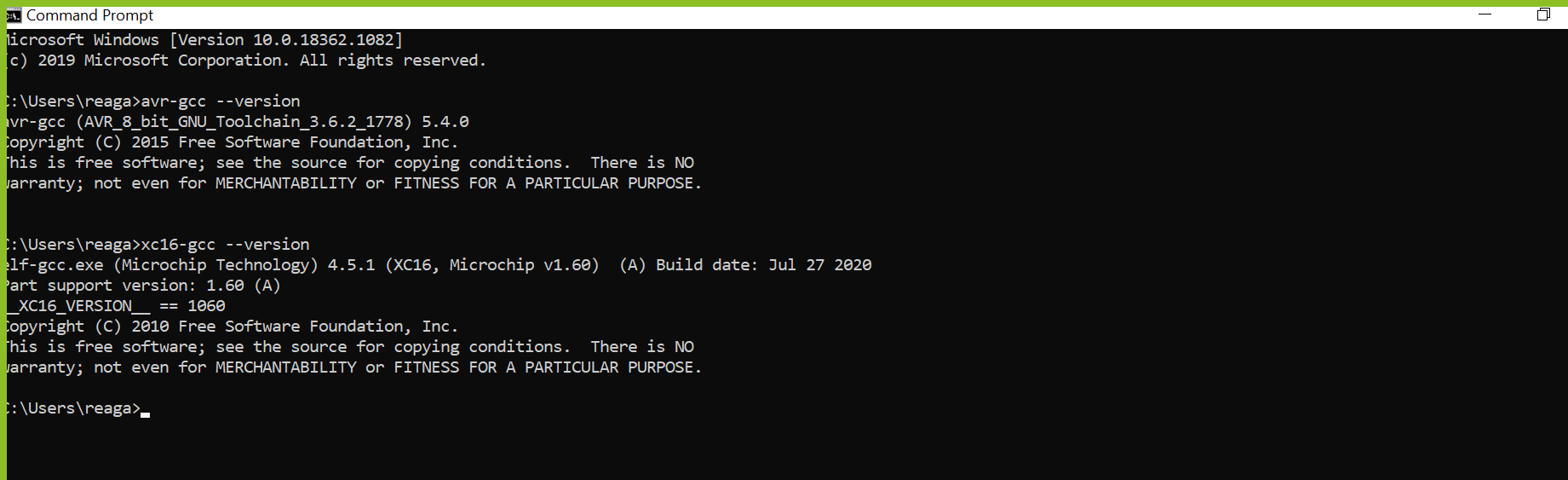
Homework 2

1. (30%)

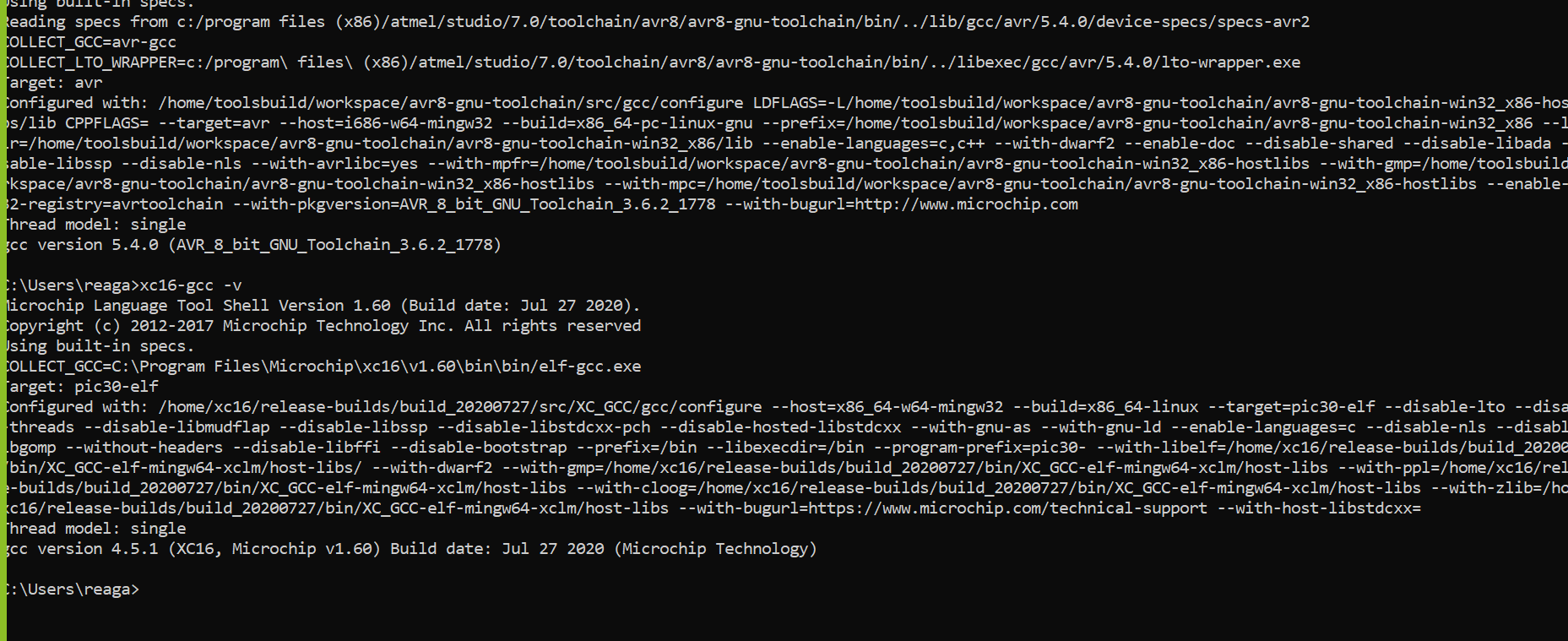
If you do not have yet, install the command-line tool chains for embedded software development, including two cross-compilers avr-gcc (Linux or Windows) and xc16-gcc (Linux, Windows, or Mac).

Report the following for the two cross-compilers respectively.

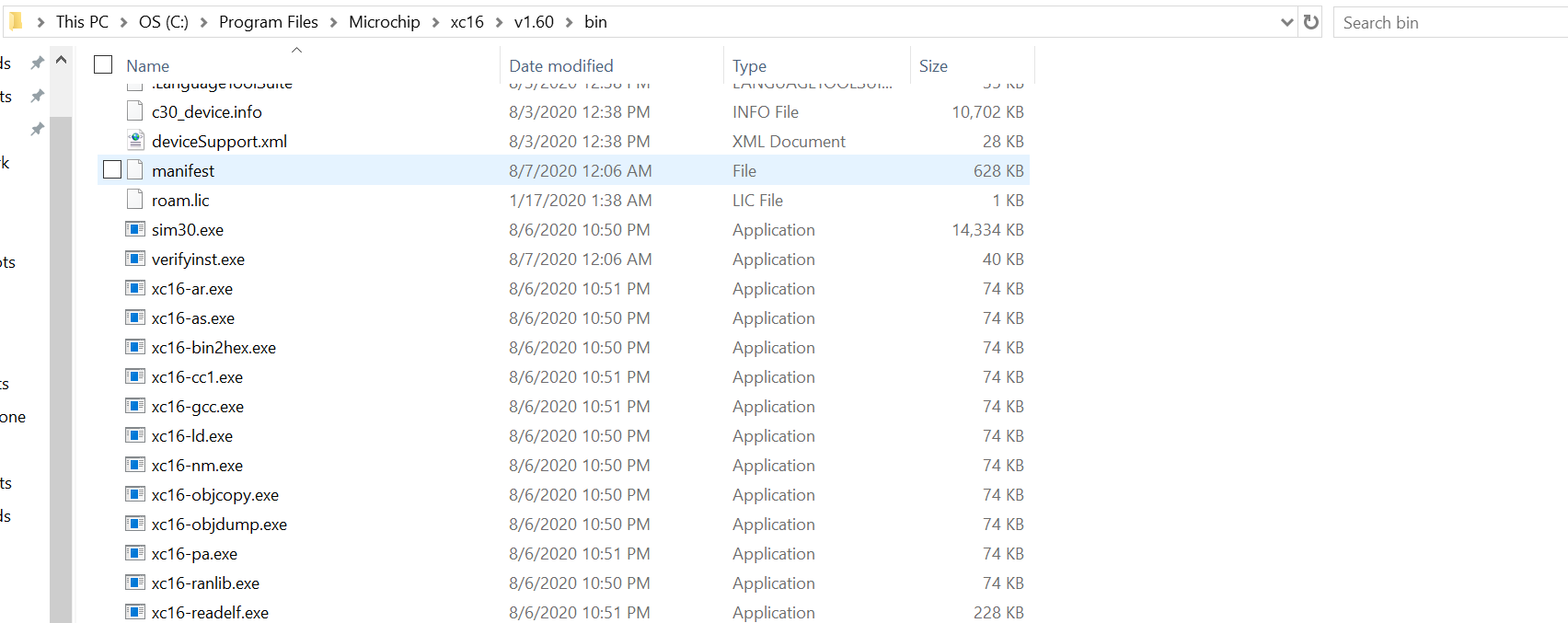
a) The version of the two compilers. Use the command like "xxx-gcc --version".

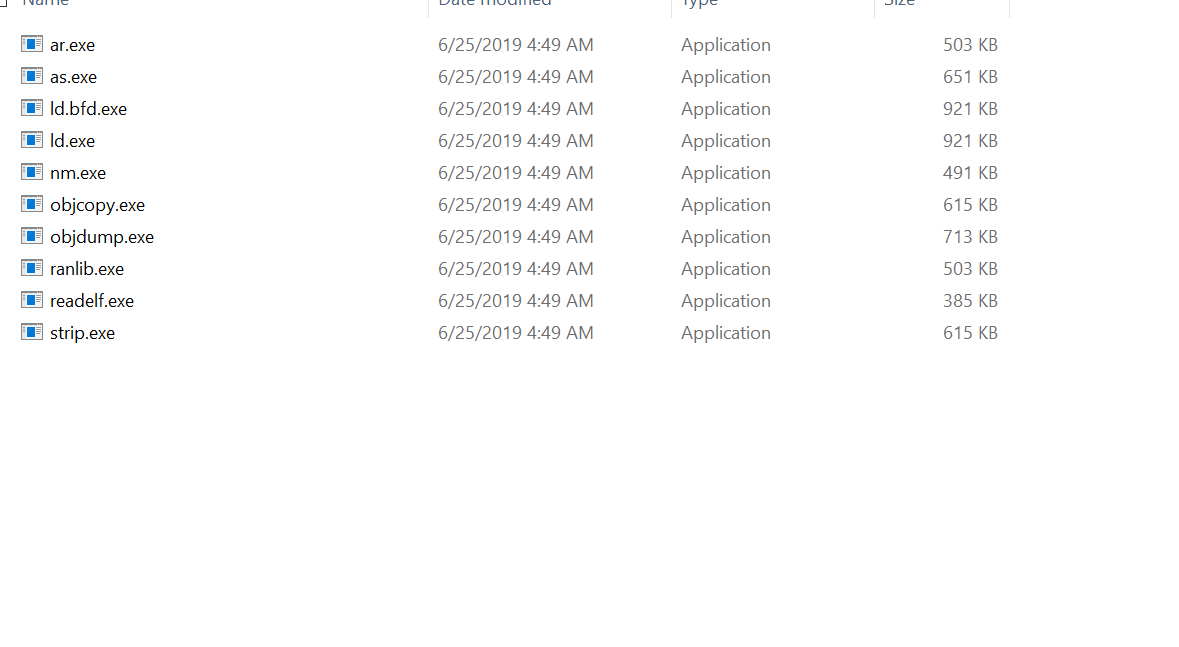


b) The folders where the two compilers are installed. You can use the command like "xxx-gcc -v".



c) The names of the avr-xxx executables and the xc16-xxx executables, for example, avr-gcc, avr-objdump and etc. At least find 5 executables.

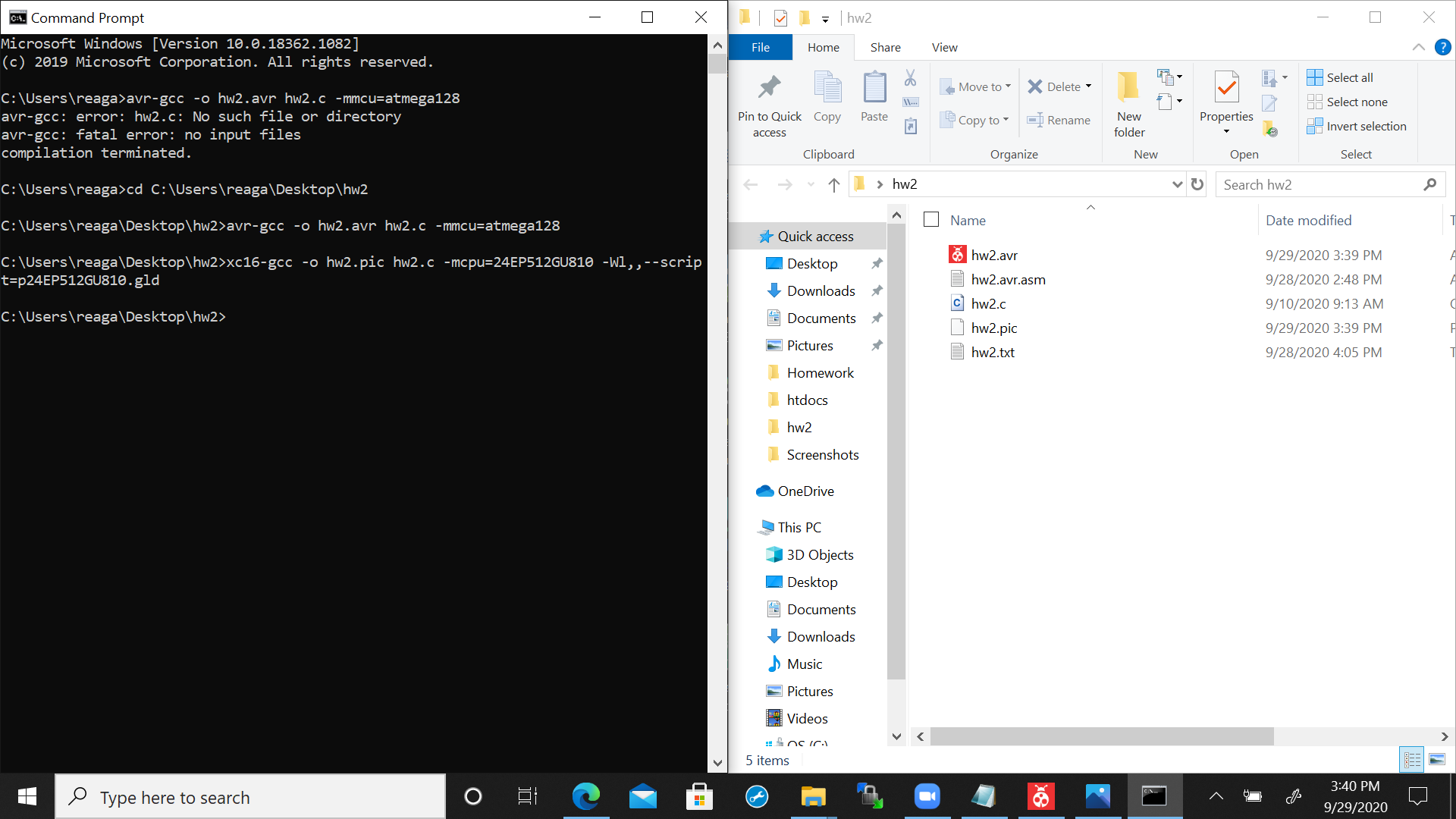




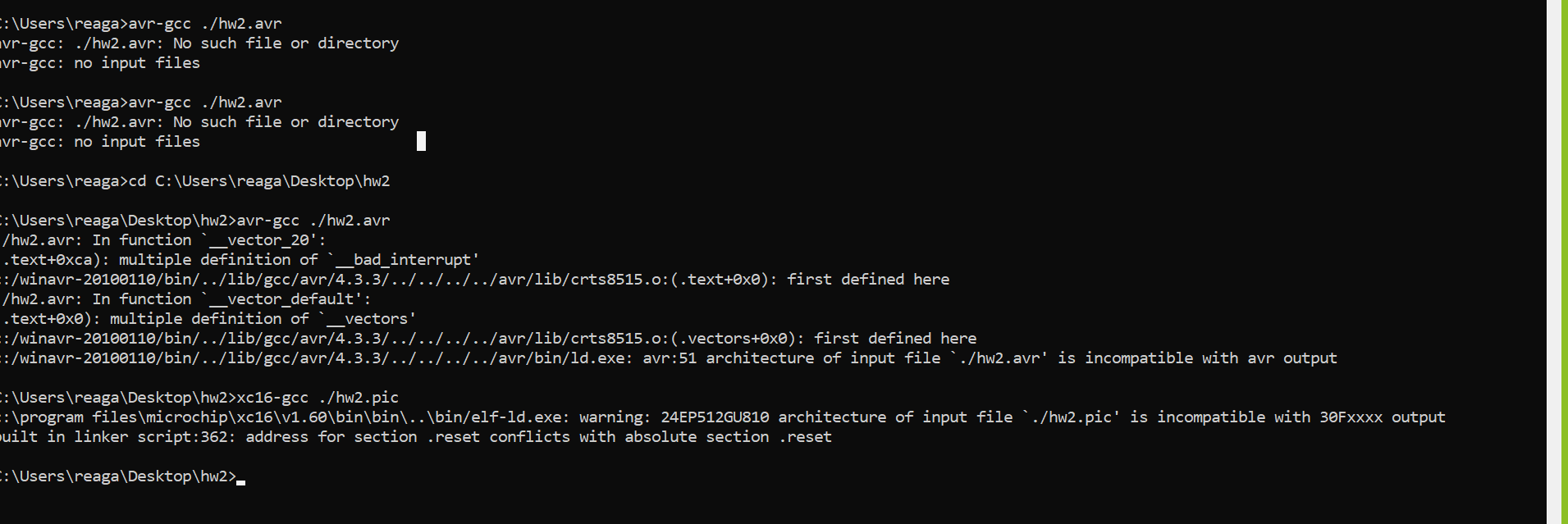
d) Compile "hw2.c" with avr-gcc and xc16-gcc and show the screenshots of compilation. The commands are.

"avr-gcc -o hw2.avr hw2.c -mmcu=atmega128"

"xc16-gcc -o hw2.pic hw2.c -mcpu=24EP512GU810 -Wl,,--script=p24EP512GU810.gld"



e) Run the two executable "hw2.avr" and "hw2.pic" in your computer and show the screenshots of the execution results.



* I was not able to run "hw2.avr" or "hw2.pic" on my computer.

2. (55%)

"hw2.avr.asm" is the AVR assembly dump of the executable built from "hw2.c". Read "hw2.avr.asm" and report the following:

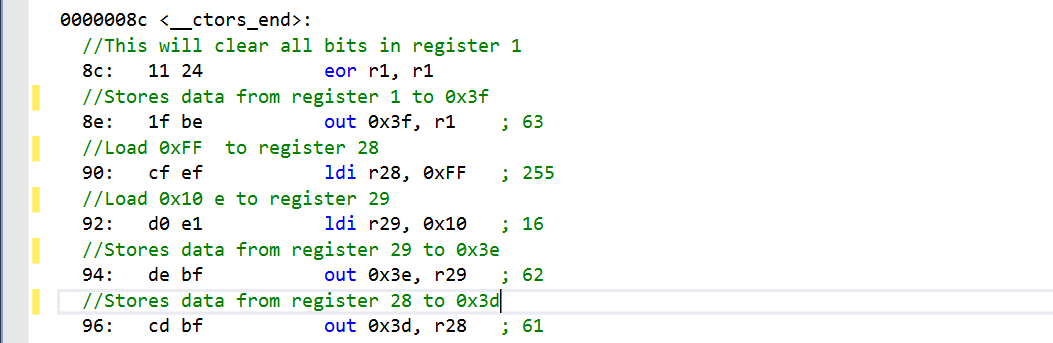
a) Convert all bytes in the .data section, <\_\_data\_start>, to ASCII chars.

* hello world!
* hello embed!

b) What is the data in the .data section and where is the data in the source code "hw2.c"?

* The data in the .data section is the bytes in the .data section and the data in the source code would be inside the main function.

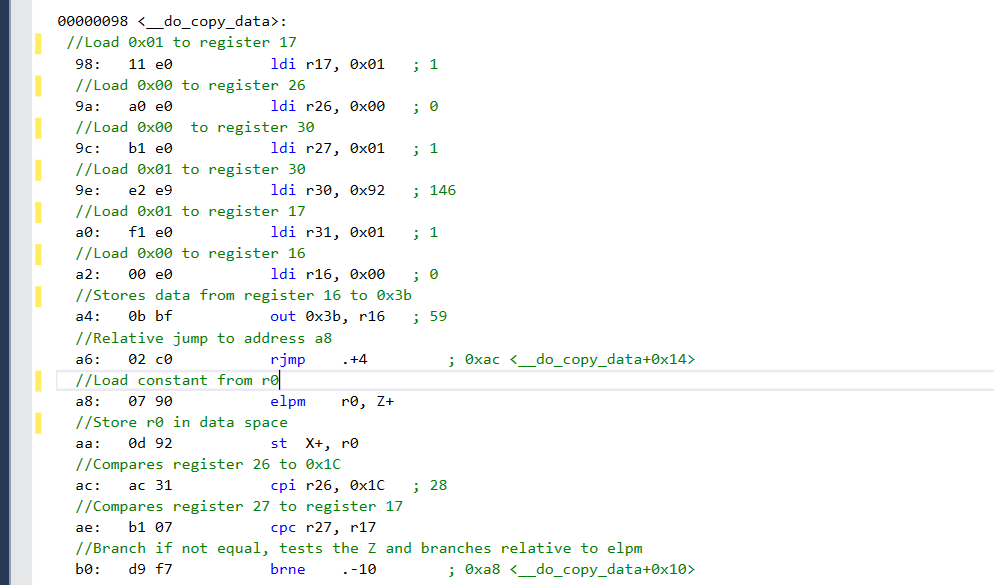
c) Comment each line of the code from line 8c to 96, i.e. the code block <\_\_ctors\_end>.



d) Explain what the code block <\_\_ctors\_end> does. What is the address of the stack bottom? Hint: IO 0x3e and 0x3d are the stack pointer.

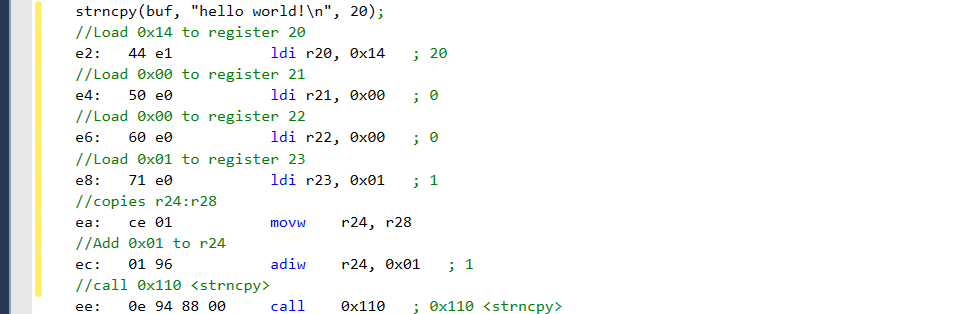
- In the code block <\_\_ctors\_end> it outputs the top of the flash memory pointer by loading and setting the register out to an I/O location that will be used later on. Based on high to low address the bottom of the stack will be register 28.

e) Comment each line of the code from line 98 to b0, i.e. the code block <\_\_do\_copy\_data>.



f) Explain what the code block <\_\_do\_copy\_data> does. How many bytes of data are copied to which address in the data memory? Hint: elpm is to load one byte from program memory (not data memory) to a register.  
 - The code block <\_\_do\_copy\_data> copies the data from the flash memory into the static RAM to be stored. The amount of bytes that were copied was 20 bytes to the 0x1C.

g) Comment each line of the code from line e2 to ee.



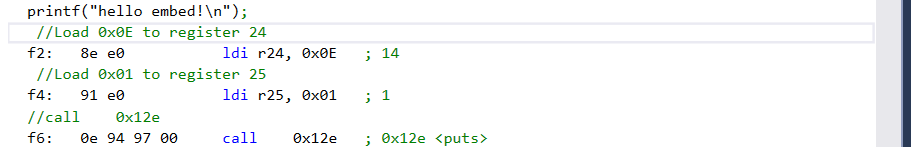
h) How are the arguments passed to calling the function strncpy? Which registers carry the first argument? Which registers carry the second argument?

* The arguments are passed to the calling function strncpy by loading the registers. The first argument is carried with register 20 and the second argument is carried with register 24.

i) How is the char array buf[] represented in AVR?

* The char array buf[] is represented in AVR with the string size of 20 since that's the number that was passed in the array. The second parameter provides the data for size and is in the source code, and will be 20 as well. The string value will not get reduced while printing.

j) Comment each line of the code from line f2 to f6.



k) How is the argument passed to calling the function puts (printf)? Which registers carry the argument?

-The argument is passed to the calling function by the registers the registers that carry the arguments are register 24 and register 25.

3. (15%)

Gcc defines integer types in stdint.h and bool type in stdbool.h.

a) Report the folders of the two header files in avr-gcc and xc16 respectively. No stdbool.h for avr-gcc.

* The two header files in avr-gcc and xc16 respectively are located in the include folder.
  + C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8\avr8-gnu-toolchain\avr\include
  + C:\Program Files\Microchip\xc16\v1.60\include

b) Report the type definitions of int8\_t, uint8\_t, int16\_t, uint16\_t, int32\_t, uint32\_t, int64\_t, uint64\_t.

Xc16:

typedef signed char int8\_t;

typedef unsigned char uint8\_t;

typedef signed int int16\_t;

typedef unsigned int uint16\_t;

typedef signed long int int32\_t;

typedef unsigned long int uint32\_t;

typedef signed long long int int64\_t;

typedef unsigned long long int uint64\_t;

Avr:

typedef signed char int8\_t;

typedef unsigned char uint8\_t;

typedef signed int int16\_t;

typedef unsigned int uint16\_t;

typedef signed long int int32\_t;

typedef unsigned long int uint32\_t;

typedef signed long long int int64\_t;

typedef unsigned long long int uint64\_t;

c) Report the type definition of bool, and the values of true and false.

