Friday questions week 1

1. To print out `hello, world` from a C program we need to "import" the declaration of the function `printf`. How do we import a file in C and which file declares `printf`?

   page 39

2. How can we ask the compiler the size in bytes of a certain type or variable?

   page 51

3. Into which four segments ("memory areas") is the address space of a running program divided?

   pages 52-54

4. A register is a type of memory which is located in the processor and is the fastest type of memory. What is the purpose of the register called the program counter?

   page 54

5. Both local variables and a function's return address are stored in a so called call frame (or activation record). Why do we need to store the return address of a function and in which of the four memory segments are the call frames stored?

   pages 54-58

6. Is a pointer simply a variable which contains a number?

   page 59

7. Consider a function declared as `void f(int a[10000]);` Calling it with an array `int x[10000]` does not result in 10000 `int`-elements being copied. Why?

   page 68

8. Assume you wish to create an array and return it to the calling function. Why is it not possible to declare the array as a variable length array or allocate memory for it using `alloca`?

   page 649

9. How would you allocate memory for an array of `n int` elements and return it to the calling function?

   page 519