

Tutorial 9

Question 1: (7)

Assuming the following variables are defined.

```
uint16_t foo = 0xA0;
uint8_t  bar = 0xF5;
```

(Answers to be given in hex)

a) What is the value of: `~foo` (1)

b) What is the value of foo, given: `foo = foo << ~bar` (2)

c) Write an equivalent two-line version of the following which clearly shows where the increment happens:

```
foo = 0xF5 - (++bar);
```

 (2)

d) After the above has executed, what are the values of `foo` and `bar`? (2)

Question 2: (5)

a) Assuming we have defined:

```
uint16_t foo = 0x0;
uint16_t *bar = &foo;
uint32_t myArray = {0, 1, 2, 3, 4};
```

a) What values will be returned by `sizeof(foo)` and `sizeof(bar)`? Explain. (2)

b) What is the value returned by: `sizeof(myArray)` (1)

Are each of the following legal or illegal? Why? (mark for reason)

c) `&foo = 0x00;` (1)

d) `bar = 0;` (1)

Question 3 (4)

a) In the context of structs, explain the difference between the member access operator and the indirect member access operator. (2)

b) Assume we declare a struct as follows:

```
struct WeirdStruct {
    uint32_t member1;
    int8_t *member2;
    int32_t member3[10];
};
```

Assume we then declare a variable as follows:

```
struct WeirdStruct ws1;
```

Assume the compiler allocates the block of memory for `ws1` starting at `0x20000150`.

What memory addresses (list all byte addresses) will the following member occupy?

```
ws1.member3[5]
```

 (2)