COMP 2401 B

Test #1 (version 3)

1. [4 marks]

   a. [3 marks]
   Answer: `0x2e + 151 = 2*16 + 14 + 151 = 46 + 151 = 197`
   `= 2^7 + 2^6 + 2^2 + 2^0 =` **`1100 0101`**
   Marking:
   -- 1 mark for correct approach
   -- 1 mark for correct addition result, given correct approach
   -- 1 mark for correct answer in binary, given correct approach

   b. [1 mark]
   Answer: **`197`**
   Marking:
   -- 1 mark for correct answer

2. [8 marks]

   a. [2 marks]
   Answer: `2^7 + 2^5 + 2^1 + 2^0 + 49 = 128 + 32 + 3 + 49 = 212`
   `212 = 2^7 + 2^6 + 2^4 + 2^2 =` **`1101 0100`**
   Marking:
   -- 1 mark for correct approach
   -- 1 mark for correct answer, given correct approach

   b. [4 marks]
   Answer: Because it's a signed char and the binary value begins with a 1, the value will be interpreted as a negative number; to get the decimal value, apply two's complement to the binary value:

	`1101 0100`
invert:	`0010 1011`
add 1:	`0010 1100`

   convert to decimal: `2^5 + 2^3 + 2^2 = 44`
   negative value that is printed: **`-44`**

   Marking:
   -- 2 marks for correctly applying two's complement
   -- 2 marks for correct negative decimal value (alt: 1 mark for positive decimal value)

c. [2 marks]
Answer: `2^7 + 2^6 + 2^4 + 2^2 =` **212**
Marking:
-- 1 mark for correct approach
-- 1 mark for correct answer, given correct approach

3. [8 marks]
Answer: **0100 0010 1011 0010 0100 0000 0000 0000**

-- sign bit: 0
-- fixed point: `89.125 = 2^6 + 2^4 + 2^3 + 2^0 + 2^-3`
`= 1011001.001 = 1.011001001 * 2^6`
-- exponent: `6 + 127 = 133 = 1000 0101`
-- fraction: `011001001`

Marking:
-- 1 mark for correct sign bit
-- 2 marks for correct fixed point representation
-- 2 marks for correct exponent in binary
-- 2 marks for correct fraction
-- 1 mark for correct final answer, padded with zeros to make 32 bits

4. [30 marks]

a. [6 marks]

```
typedef struct {                    // 1 mark for typedef struct
  char species[MAX_STR_SIZE];       // 1 mark for species
  char name[MAX_STR_SIZE];          // 1 mark for name
  int  age;                         // 1 mark for age
  char gender;                      // 1 mark for gender
} AnimalType;                       // 1 mark AnimalType
```

b. [8 marks]
i. [2 marks] Answer: output
ii. [2 marks] Answer: input
iii. [2 marks] Answer: input
iv. [2 marks] Answer: input-output

c. [16 marks]

```
// 1 mark for looping over correct collection
  for (int i=0; i<org->size; ++i) {

// 8 marks for correct condition
// -- 4 marks for correctly comparing species
// -- 2 marks for correctly comparing gender
// -- 2 marks for correctly comparing age
    if ( strcmp(org->animals[i].species, species) == 0
     && org->animals[i].gender == gender
     && org->animals[i].age <= age ) {

// OPTION #1:
// 5 marks for correctly adding to sel
// -- 2 marks for assigning from org current element
// -- 2 marks for assigning to sel
// -- 1 mark for assigning to end of sel
      sel->animals[sel->size] = org->animals[i];
// 2 marks for incrementing sel size
      sel->size++;

// OPTION #2:
// 3 marks for calling addAnimal
// 2 marks for using sel as param
// 2 marks for using org current element
      addAnimal(sel, &(org->animals[i]));
    }
  }
```