COMP 2401 B

Test #1 (version 1)

1. [4 marks]

   a. [3 marks]
   Answer: `0x3c + 105 = 3*16 + 12 + 105 = 60 + 105 = 165`
   `= 2^7 + 2^5 + 2^2 + 2^0 =` **`1010 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: **`165`**
   Marking:
   -- 1 mark for correct answer

2. [8 marks]

   a. [2 marks]
   Answer: `2^7 + 2^4 + 2^3 + 2^2 + 2^0 + 46 = 128 + 16 + 13 + 46 = 203`
   `203 = 2^7 + 2^6 + 2^3 + 2^1 + 2^0 =` **`1100 1011`**
   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:

	`1100 1011`
invert:	`0011 0100`
add 1:	`0011 0101`

   convert to decimal: `2^5 + 2^4 + 2^2 + 2^0 = 53`
   negative value that is printed: **`-53`**

   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 + 2^3 + 2^1 + 2^0 =` **203**
Marking:
-- 1 mark for correct approach
-- 1 mark for correct answer, given correct approach

3. [8 marks]
Answer: **0100 0010 0001 0101 1000 0000 0000 0000**
-- sign bit: 0
-- fixed point: `37.375 = 2^5 + 2^2 + 2^0 + 2^-2 + 2^-3`
`= 100101.011 = 1.00101011 * 2^5`
-- exponent: `5 + 127 = 132 = 1000 0100`
-- fraction: `00101011`

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-output
iii. [2 marks] Answer: input
iv. [2 marks] Answer: input

c. [16 marks]

```
  int index;

// 6 marks for finding the correct animal
// -- 2 marks for correct loop and break
// -- 2 marks for correctly using strcmp
// -- 1 mark for comparing animal name at current array element
// -- 1 mark for comparing name passed in as parameter

  for (index=0; index<arr->size; ++index) {
    if (strcmp(arr->animals[index].name, name) == 0)
      break;
  }

// 2 marks for checking for name not found

  if (index == arr->size)
    return;

// 6 marks for shifting elements toward the front of array
// -- 2 marks for correct loop header, from index to end
// -- 4 marks for correctly shifting elements

  for (int i=index; i<arr->size-1; ++i) {
    arr->animals[i] = arr->animals[i+1];
  }

// 2 marks for decrementing array size
  arr->size--;
```