

Quiz 1

CMSC 14300 - Systems Programming
Covers Lectures 1 through 4

Instructions. Closed book, no computer, about 25 minutes. Write your C by hand. You do not need to write `#include` lines or a full `main` unless a problem asks for one. Minor syntax slips are fine as long as your logic is clear. A reference sheet of conversion tables is on the last page. You may use a single letter-sized cheat sheet for the quiz.

Name: _____

Total: 25 points

Part A. Short answer (8 points)

A1. (4 pts; 1 pt each) Give the value of each expression. Assume any operands are `int` (or `unsigned int` for the bitwise ones).

(a) `20 % 6`

(b) `2 + 3 * 4`

(c) `6 ^ 3`

(d) `0xF0 >> 4`

A2. (4 pts; 1 pt each) Number bases and operators.

(a) Convert the decimal value 76 to 8-bit binary.

(b) Convert the binary value 1011 0010 to hexadecimal.

(c) What decimal value is the hex literal 0x5A?

(d) If `n` is 0, does evaluating `(n != 0 && 100 / n > 5)` cause a divide-by-zero? Answer yes or no, and explain in one sentence.

Part B. Reading code (8 points)

B1. (5 pts) What does this program print? Write the exact output, including spaces, on one line.

```
1 #include <stdio.h>
2
3 int main(void) {
4     int x = 2;
5     switch (x) {
6         case 1: printf("one ");
7         case 2: printf("two ");
8         case 3: printf("three "); break;
9         case 4: printf("four ");
10        default: printf("done");
11    }
12    printf("\n");
13    return 0;
14 }
```

B2. (3 pts) This function is supposed to return the sum of the `n` elements of the array `a`. It has **two** bugs. Identify both and give the corrected lines.

```
1 int sum(int a[], int n) {
2     int total;
3     for (int i = 0; i <= n; i++) {
4         total += a[i];
5     }
6     return total;
7 }
```

Part C. Bit packing: the monster stat word (9 points)

A handheld monster-battling game (whose name shall be left out for copyright reasons) stores each creature's stats packed into a single **unsigned int** (Assume that it is **16 bits**; bits 14 and 15 are unused and always 0). The layout is:

Field	Bit positions	Width	Range
Level	0–3	4 bits	0–15
HP	4–9	6 bits	0–63
Type	10–12	3 bits	0–7
Shiny (S)	13	1 bit	0–1
unused	14–15	2 bits	always 0

A picture of the 16 bits (most significant on the left):

bit:	15	14		13		12	11	10		9	8	7	6	5	4		3	2	1	0
field:	unused			S		Type				HP							Level			

C1. (5 pts) A creature that looks very similar to (but is definitely not) Charizard, has **Level = 12**, **HP = 40**, **Type = 5**, and is **Shiny** (Shiny = 1). Pack these into the 16-bit stat word.

- (a) (3 pts) Write the full 16-bit value in binary, grouped into nibbles.
- (b) (2 pts) Write that value in hexadecimal.

First encode each field to binary:

Field	Value (dec)	Width	Binary
Level	12	4 bits	
HP	40	6 bits	
Type	5	3 bits	
Shiny	1	1 bit	
unused	0	2 bits	00

- (a) Now place each field in its bit positions:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- (b) In hexadecimal: 0x

C2. (4 pts; 1 pt each) A saved monster is stored as the stat word 0x0D4A. Decode all four fields.

First decode 0x0D4A to its 16 bits:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Now read off each field's bits and convert to decimal:

Field	Bit positions	Binary	Value (dec)
Level	0–3		
HP	4–9		
Type	10–12		
Shiny	13		

End of Quiz 1.

Reference sheet

These tables are provided for your convenience. You do not need to memorize them.

Decimal / Binary / Hex (0 to 15)

Dec	Binary	Hex
0	0000	0x0
1	0001	0x1
2	0010	0x2
3	0011	0x3
4	0100	0x4
5	0101	0x5
6	0110	0x6
7	0111	0x7
8	1000	0x8
9	1001	0x9
10	1010	0xA
11	1011	0xB
12	1100	0xC
13	1101	0xD
14	1110	0xE
15	1111	0xF

Powers of two

2 ⁿ	value
2 ⁰	1
2 ¹	2
2 ²	4
2 ³	8
2 ⁴	16
2 ⁵	32
2 ⁶	64
2 ⁷	128

2 ⁿ	value
2 ⁸	256
2 ⁹	512
2 ¹⁰	1024
2 ¹¹	2048
2 ¹²	4096
2 ¹³	8192
2 ¹⁴	16384
2 ¹⁵	32768

Handy facts: $x \ll n$ multiplies by 2^n ; $x \gg n$ divides an unsigned value by 2^n . A field of width w has 2^w possible values and a mask of w ones (for example, 6 ones is `0x3F`).