CMPU-224 Exam 1
Spring 2026

Name: _____

This is a closed book, closed notes quiz. No electronic devices are allowed. You have 75 minutes to complete the quiz. There are a total of 13 questions and 100 points. This quiz has 10 pages. Make sure you have all 10 pages before starting the quiz.

There should be enough space on the quiz for your answers. If you need more space to work out a problem, blank paper will be available, just ask.

Good Luck!

1. (6 points) You are given the following dump of memory below. The format of the memory dump is `<address>:<value>`. For example, the byte at memory address `0x6f5` is `0x61` and the byte at memory address `0x70a` is `0xc9`.

```
6f0:81 6f1:d5 6f2:14 6f3:2f 6f4:e5 6f5:61 6f6:0c 6f7:86 6f8:18 6f9:74 6fa:49 6fb:08 6fc:57 6fd:f0 6fe:c3 6ff:91
700:e3 701:55 702:32 703:bb 704:ae 705:62 706:3a 707:69 708:f9 709:71 70a:c9 70b:9f 70c:ab 70d:01 70e:97 70f:cb
710:a9 711:8d 712:9b 713:3e 714:fb 715:29 716:87 717:21 718:7b 719:35 71a:43 71b:2e 71c:42 71d:f3 71e:17 71f:39
720:84 721:9a 722:ce 723:9c 724:67 725:3f 726:16 727:4f 728:56 729:44 72a:ec 72b:d9 72c:e6 72d:92 72e:eb 72f:8c
730:a4 731:6c 732:23 733:d6 734:65 735:26 736:94 737:b0 738:ca 739:68 73a:a6 73b:1a 73c:ea 73d:53 73e:25 73f:f6
740:59 741:f4 742:96 743:47 744:6f 745:1f 746:a0 747:b5 748:be 749:64 74a:bf 74b:c6 74c:40 74d:a3 74e:63 74f:77
750:b2 751:dc 752:8b 753:0b 754:2a 755:78 756:0a 757:dd 758:5d 759:46 75a:50 75b:fd 75c:a8 75d:11 75e:d7 75f:7a
760:7e 761:ad 762:cd 763:02 764:58 765:2b 766:2d 767:4d 768:b6 769:73 76a:98 76b:de 76c:b3 76d:88 76e:1e 76f:09
770:3b 771:d0 772:85 773:4b 774:31 775:24 776:a1 777:f5 778:f2 779:d3 77a:a7 77b:fc 77c:28 77d:db 77e:04 77f:cf
780:2c 781:76 782:c5 783:05 784:c7 785:6a 786:7f 787:af 788:3d 789:5f 78a:a5 78b:60 78c:0e 78d:e4 78e:8a 78f:75
```

What are the values of the following expressions below? **Give your answers in hexadecimal**.

Expression	**Value**
Little-endian **`char`** at address **`0x723`**	0x
Big-endian **`char`** at address **`0x715`**	0x
Little-endian **`short`** at address **`0x738`**	0x
Big-endian **`short`** at address **`0x72e`**	0x
Little-endian **`int`** at address **`0x740`**	0x
Big-endian **`int`** at address **`0x73c`**	0x

2. (3 points) Convert the base-4 number 213_4 to base-10: _____

3. Convert the following 8-bit two's complement numbers to 16-bit two's complement numbers. Give your answers in **hexadecimal**.

   (a) (3 points) `0xB4`: _____

   (b) (3 points) `0x3A`: _____

4. What are the smallest (most negative) and largest (most positive) 15-bit two's complement numbers? Give your answers in **hexadecimal**.

   (a) (3 points) Smallest 15-bit number: _____

   (b) (3 points) Largest 15-bit number: _____

5. (8 points) Write the following function, `bool_bits()`.

```
/*
 * bool_bits - return 1 if any of the bits of byte 2 (bits 16 to 23)
 * of the input are one.  Bits are numbered from 0 (least significant) to 31
 * (most significant). The value of the other bits do not matter for this function.
 *
 *   Examples: bool_bits(0x00ff0000) = 1
 *             bool_bits(0xff00ffff) = 0
 *             bool_bits(0x00100000) = 1
 *
 */
```

Legal operations:

```
! ~ & ^ | + << >>
```

and any 1-byte (0 to 255) numeric constants.

```
int bool_bits(int x) {









}
```

6. Recall that floating point numbers are represented in the form $V = (-1)^s * M * 2^E$ where M is encoded in `frac`, and E is encoded in `exp`. Assume we have the following tiny **12-bit** floating point representation where bits 0-6 represent the `frac` field (7 bits), bits 7-10 represent the `exp` field (4 bits), and bit 11 represents the sign bit (1 bit).

   (a) (3 points) What is the value of positive infinity ($+\infty$) in the floating point system described above?

   Give your answer in hexadecimal. 0x_____

   (b) (3 points) In this system, which of the following is a representation of a negative NaN? (select one).

   ◯ `0x780`
   ◯ `0xF80`
   ◯ `0xF81`
   ◯ `0x081`

   (c) (3 points) In the above system, which of the following is a representation of a negative denormalized number? (select one)

   ◯ `0x900`
   ◯ `0x002`
   ◯ `0x882`
   ◯ `0x802`

(d) (3 points) What is the value of decimal fraction `13.4` in the floating point system described above?

Give your answer in hexadecimal. 0x_____

(e) (3 points) You are given the following hexadecimal number `0x4A4` in the floating point system described above. What is the decimal representation of that number?

Give your answer as a **decimal fraction**, representing the fractional part of the number (if it has one) as a fraction (e.g., 3/4).

_____

7. Round the following binary fractions to exactly 3 fractional bits using the "round to even" rounding rules.

(a) (2 points) $1.01101_2 \rightarrow$ _____

(b) (2 points) $1.01011_2 \rightarrow$ _____

(c) (2 points) $1.11010_2 \rightarrow$ _____

(d) (2 points) $1.10110_2 \rightarrow$ _____

8. (16 points) For each of the following assembly code snippets, determine the final decimal value stored in register `a0`.

Prior to the execution of *each* snippet, assume the following initial machine state:

Registers:

- `t0 = 16`
- `t1 = 32`
- `t2 = -4`
- `t3 = 0x100`
- `x0 = 0` (always)

Memory (Little-Endian):

Address	Value (32-bit Word)
`0x100`	`0x00000005` (5)
`0x104`	`0x0000000A` (10)
`0x108`	`0xFFFFFFFF` (-1)
`0x10C`	`0x00000000` (0)

Give your answers in **decimal**.

(a)
```
addi a0, t3, 4
lw a0, 4(a0)
```

a0 is _____ (in decimal)

(b)
```
srai a0, t2, 2
slli a0, a0, 1
```

a0 is _____ (in decimal)

(c)
```
lh a0, 8(t3)
addi a0, a0, 5
```

a0 is _____ (in decimal)

(d)
```
lui a0, 2
srli a0, a0, 11
```

a0 is _____ (in decimal)

(e)
```
lbu a0, 8(t3)
addi a0, a0, 1
```

a0 is _____ (in decimal)

(f)
```
sltu a0, t2, t1
xori a0, a0, 1
```

a0 is _____ (in decimal)

(g)
```
lw a0, 4(t3)
ori a0, a0, 5
```

a0 is _____ (in decimal)

(h)
```
mul a0, t0, t2
sub a0, a0, t1
```

a0 is _____ (in decimal)

9. Consider the following structure declaration, assuming a 32-bit architecture (RV32) where pointers are 4 bytes and doubles are 8 bytes:

```
struct record {
    short   id;
    double  score;
    char    *name;
    char    grade;
    int     rank;
};
```

(a) (2 points) Give the total size of the structure in bytes (i.e., the value of the `sizeof(struct record)` expression). Include any required padding.

`sizeof(struct record)` returns _____

(b) (2 points) Assume you have the following C code utilizing the above structure:

```
long get_rank(struct record *x) {
    return x->rank;
}
```

Fill in the missing blanks below for the assembly language implementation of the above function.

```
get_rank:



    ret
```

(c) (2 points) If the structure declaration above was rearranged to minimize the space of the structure, what would be the new size of the structure?

Total size of rearranged and minimized `struct` in bytes: _____

10. (4 points) You have the following two-dimensional array of 16-bit integers (shorts) declared as follows:

```
short grid[8][6];
```

Assume the start of the array is at address `0x4000`. You have an array access `grid[x][y]`. What are the indices (`x` and `y` below) of the array that refers to the array element at address `0x403A`?

`x is` _____

`y is` _____

11. (16 points) You are given the following RISC-V assembly and C code. Fill in the missing bits of the C program. **Note**: each blank should be either a single number, variable, or operation.

RISC-V assembly

```
question:
   101d0: mv   a5,a0
   101d4: li   a0,0
   101d8: j    101e4 <question+0x14>
   101dc: addi a5,a5,4
   101e0: addi a1,a1,-1
   101e4: blez a1,10200 <question+0x30>
   101e8: lw   a4,0(a5)
   101ec: andi a3,a4,1
   101f0: bnez a3,101dc <question+0xc>
   101f4: bge  a2,a4,101dc <question+0xc>
   101f8: addi a0,a0,1
   101fc: j     101dc <question+0xc>
   10200: ret
```

C code

```
int question(int a[], int len, int threshold) {

    int count = 0;

    int val;

    while (________ > ________) {

        ________ = *a;

        if ((val ________ 1) == 0) {

            if (val ________ threshold) {

                ________ = count + 1;

            }

        }

        a = a + ________;

        len = len - 1;

    }

    return ________;

}
```

12. (3 points) How does a stack canary mitigate stack smashing attacks?

- ◯ By placing a randomized, secret value between local variables and the return address, and verifying it hasn't changed before returning
- ◯ By marking the stack memory region as non-executable, preventing any injected shellcode from running
- ◯ By randomizing the memory locations of the stack, heap, and libraries every time the program runs
- ◯ By statically analyzing the code at compile time to ensure no array bounds are violated

13. (3 points) Return-Oriented Programming (ROP) is an exploit technique primarily designed to bypass which of the following security defenses?

- ◯ Address Space Layout Randomization (ASLR)
- ◯ Stack canaries
- ◯ Non-executable memory protections
- ◯ Control Flow Integrity (CFI)