1. Suppose you want to encode (represent in binary) the 52 playing cards in a standard deck of cards. What is the *minimum* number of bits you would need? Your answer: _____.

2. What *decimal* value does **`1011`** represent? Give the answer for each of the following representation systems. Assume each system uses 4 bits.

	your answer
• unsigned binary	_____
• one's complement	_____
• two's complement	_____
• sign and magnitude	_____

3. Represent the *decimal value* **`-4`** in each of the following systems. Assume 4 bits. Answers using fewer or more than 4 bits will not be considered correct. If it's not possible to represent **`-4`** in a given system, write NR (for Not Representable).

	your answer
• unsigned binary	_____
• one's complement	_____
• two's complement	_____
• sign and magnitude	_____

4. Convert the hex number **`CAFE`** to binary.

   Your answer: _____.

5. Convert the binary number **`111010.10101`** to hex.

   Your answer: _____.

6. *(4 points)* Give the range of values (in *decimal*) that 6 bits is capable of representing using *two's complement*. Your answer: _____.

7. Using *Horner's method*, find the *decimal* value corresponding to the *unsigned* binary number **`100110101`**.
   IMPORTANT: You must show working that clearly demonstrates the use of Horner's method to get any points.

8. Using *Horner's method*, find the *decimal* value corresponding to the *two's complement* number **`100110101`**.
   IMPORTANT: You must show working that clearly demonstrates the use of Horner's method to get any points.

9. Perform binary addition for the following pair of numbers in *two's complement*. Circle Y or N to indicate if overflow has occurred and *describe how you arrived at your answer* concerning overflow.

```
    10101001
    11010110
(+) --------
```

overflow occurred: [Y , N]

10. Represent the decimal value **-12.5625** in IEEE 754-1985 single precision.

HINTS: $(-1)^{S} \times (1 + \text{Fraction}) \times 2^{(\text{Exponent} - 127)}$

S	Exponent (8 bits)	Fraction (23 bits)

Your answer: _____.
(To be eligible for potential partial credit, show working below.)

11. Suppose the 32-bit hex value **`ABCD4321`** has been stored in memory starting at address 1000 (in decimal). Assuming byte-addressable memory, show the contents of the following memory locations if the machine is *little endian* and if it is *big endian*.

address (in decimal)	Contents in hex (if *little endian*)	Contents in hex (if *big endian*)
1000		
1001		
1002		
1003		

12. In the ASCII character set, the binary encoding of an uppercase letter and that of the corresponding lowercase letter differs by only 1 bit, which is the _____ bit.

    NOTE: Let's refer to the least significant bit (LSB) as the 1st bit.

    HINT: ASCII codes for **`'A'`** and **`'c'`** (in hex) are **`41`** and **`63`**, respectively.

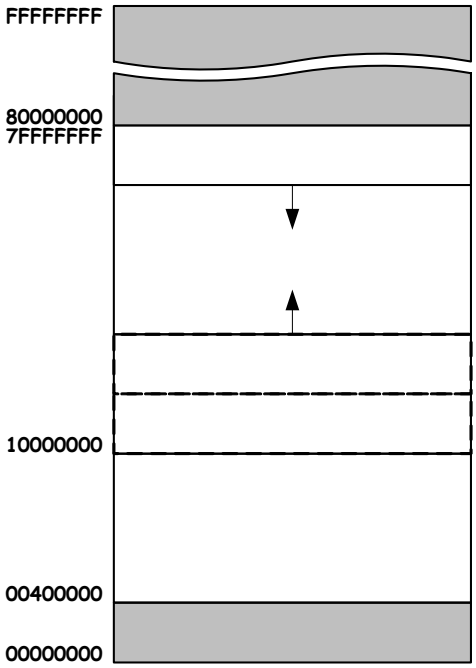
13. Give 3 key characteristics of the von Neumann architecture.

    - _____
    - _____
    - _____

14. In a *load-store* architecture, ...

    ☐ The instruction set consists of only load and store instructions.
    ☐ Every instruction in the instruction set has an operand that is a memory location.
    ☐ In the instruction set, the only instructions that access memory are load and store instructions.
    ☐ None of the above.

15. The diagram shown on the right depicts the typical memory layout for the MIPS32 architecture. You are to name the main segments (3 of them) and sub-segments (2 of them) discussed in class and show where they are located in the diagram.



16. According to the usage convention for MIP's CPU registers, …

- ☐ The *calling function* is responsible for preserving the contents of *caller-saved* registers and the *called function* for preserving *callee-saved* registers.
- ☐ The *called function* is responsible for preserving the contents of *caller-saved* registers and the *calling function* for preserving *callee-saved* registers.

17. [**T** or **F**]The **addu** and **subu** instructions treat their operands as *unsigned* integers.

18. Why is the width of every *register specifier field* in a MIPS32 instruction 5 bits?

19. Write MIPS assembly code segment to do the "equivalent" of what the following C++ statement does: **`int intArray[] = {1, 2, 3, 4, 5};`**

20. Write a simple but complete MIPS assembly language program to do the following:
    - Ask the user to enter two integers with the prompt **`"Enter 2 integers: "`**.
    - Read the two integers entered.
    - Compute the product of the two integers. (Assume the product won't exceed 32 bits.)
    - Display the result in the format **`"28 is the product of 7 and 4"`**.