Devising a way to storing **unsigned** numbers inside a computer

- **Computers** use the **"binary number system" positional system** to represent **unsigned numbers**.

- The **Binary number system** is **similar** to the **decimal number system**:
  - **Decimal number system:**
    - **10 digits** are used to represent the following **values**:

      ```
      digit      value
      --------------------------
        0    =    ( )
        1    =    (*)
        2    =    (**)
        3    =    (***)
        4    =    (****)
        5    =    (*****)
        6    =    (******)
        7    =    (*******)
        8    =    (********)
        9    =    (*********)
      ```

    - A **digit** that appears at the ***n*-th position** to the **left** of the **decimal point** has a **value** equal to:

      value of the digit × 10^n

- **Binary number system**:
  - **Binary number system:**
    - **2 digits** are used to represent the following **values**:

      ```
      digit      value
      --------------------------
        0    =    ( )
        1    =    (*)
      ```

    - A **digit** that appears at the ***n*-th position** to the **left** of the **decimal point** has a **value** equal to:

      value of the digit × 2^n

- **Example:**

```
Unsigned value    4 bit binary number    8 bit binary number
--------------    -------------------    -------------------
       0                0000                 00000000
       1                0001                 00000001
       2                0010                 00000010
       3                0011                 00000011
```

Converting between values and their binary representations

- Given a **binary number representing** an unsigned value, how to **determine the value represented** ?
  - **Answer:**
    - **Sum** the value of each digit **multiplied** by its **"location factor"**
  - **Example:**
    - What **unsigned value** is represented by **01011001** ?

- **Answer:**

```
Digits:            0   1   0   1   1   0   0   1
Location factor:   128 64  32  16  8   4   2   1
                  --------------------------------
                       64  +   16 +8          + 1 = 89

Answer: 89
```

- Given an **unsigned value**, how to find the **binary number** that represent that value ?
  - **Answer:**
    - **Divide** the **value** repeatedly by **2** and
    - **Collect** the **remainders** of each **division** in the **reverse order**.
  - **Example:**
    - What is the **8 bit binary number** that represents the unsigned value **89** ?
    - **Answer:**

```
                89
           /2 ------ 1   <--- remainder of 89/2
                44       <--- quotient of 89/2
           /2 ------ 0   <--- remainder of 44/2
                22       <--- quotient of 44/2
           /2 ------ 0
                11
           /2 ------ 1
                 5
           /2 ------ 1
                 2
           /2 ------ 0
                 1
           /2 ------ 1
                 0      <----- Done

Binary number: 1011001 (7 bits)

Pad to 8 bits: 01011001

Answer: 01011001
```

- **Storing binary representations of unsigned values inside a computer**
  - **Facts:**
    - We can **manufacture electronic *switches***
      - A **switch** can be **on** or **off**
      - The **on *state*** represents a **1 bit**
      - The **off *state*** represents a **0 bit**
    - A **byte (memory)** = **8 switches**
  - **Storing** the **representation** of **unsigned values**:
    - **Unsigned values** are stored in **memory** using the **binary number system**

      Example:
      - How to **represent** the value **five (5)** using **1 byte memory (= byte variable)**:

```
00000101
```

- How to **represent** the value **five (5)** using **2 bytes memory (= short variable)**:

  ```
  00000000 00000101
  ```

- How to **represent** the value **five (5)** using **4 bytes memory (= int variable)**:

  ```
  00000000 00000000 00000000 00000101
  ```

- **Quiz:**

  - What **bit pattern** is stored in memory (= in the memory cells reserved for the variable) by the following **Java program**:

    ```
    int  x;

    x = 9;
    ```

  - What **bit pattern** is stored in memory (= in the memory cells reserved for the variable) by the following **Java program**:

    ```
    short  x;

    x = 9;
    ```

  - What **bit pattern** is stored in memory (= in the memory cells reserved for the variable)) by the following **Java program**:

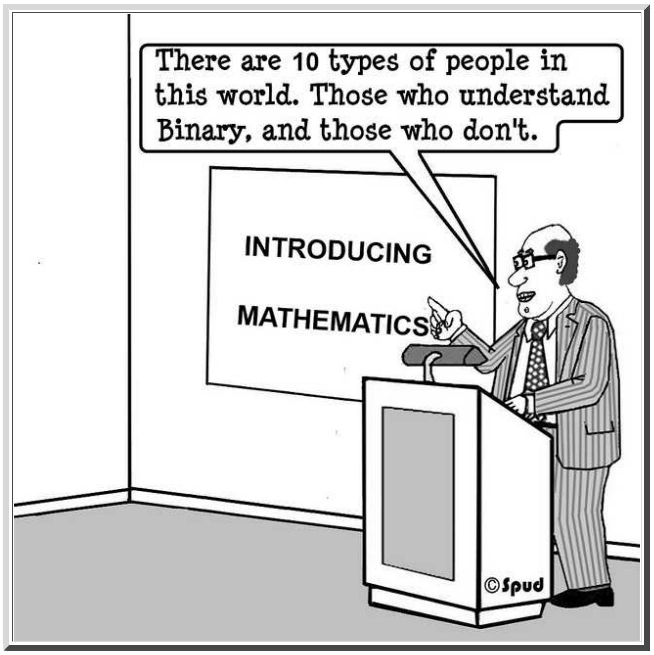
    ```
    byte  x;

    x = 9;
    ```

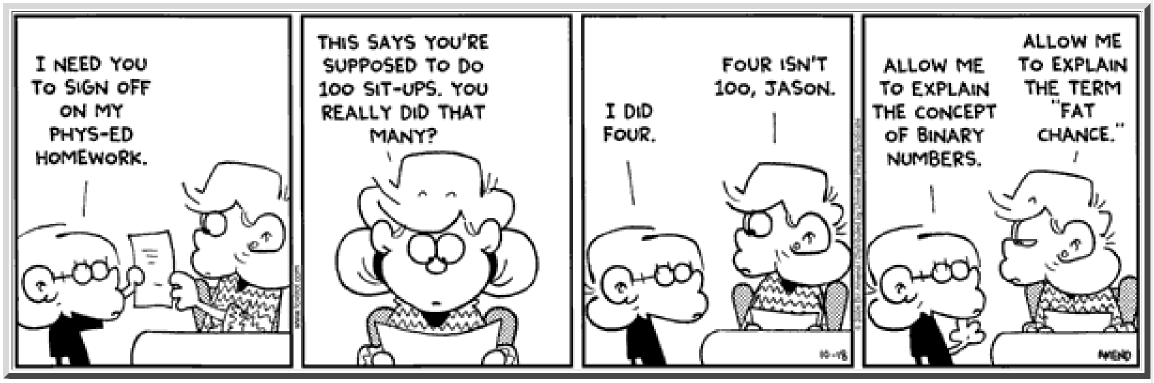
- **Test yourself if you understand binary numbers....**

  - See if you understand the following jokes:

    - Joke #1:


There are 10 types of people in this world. Those who understand Binary, and those who don't.
INTRODUCING
MATHEMATICS
©Spud

- Joke #2:


I NEED YOU TO SIGN OFF ON MY PHYS-ED HOMEWORK.
THIS SAYS YOU'RE SUPPOSED TO DO 100 SIT-UPS. YOU REALLY DID THAT MANY?
I DID FOUR.
FOUR ISN'T 100, JASON.
ALLOW ME TO EXPLAIN THE CONCEPT OF BINARY NUMBERS.
ALLOW ME TO EXPLAIN THE TERM "FAT CHANCE."
10-18
AMEND