CMSC 11:
Introduction to Computer Science

Jaderick P. Pabico <jppabico@uplb.edu.ph>
Institute of Computer Science, CAS, UPLB

Review

- Memory
  - Electronic vs. Electromechanical
  - Internal Memory
  - Memory Addressing
  - Maximum Number of Addressable Cells
    - Example: 32-bit, 8-bit
- Hexadecimal
- Types of Internal Memory: Core, RAM, ROM

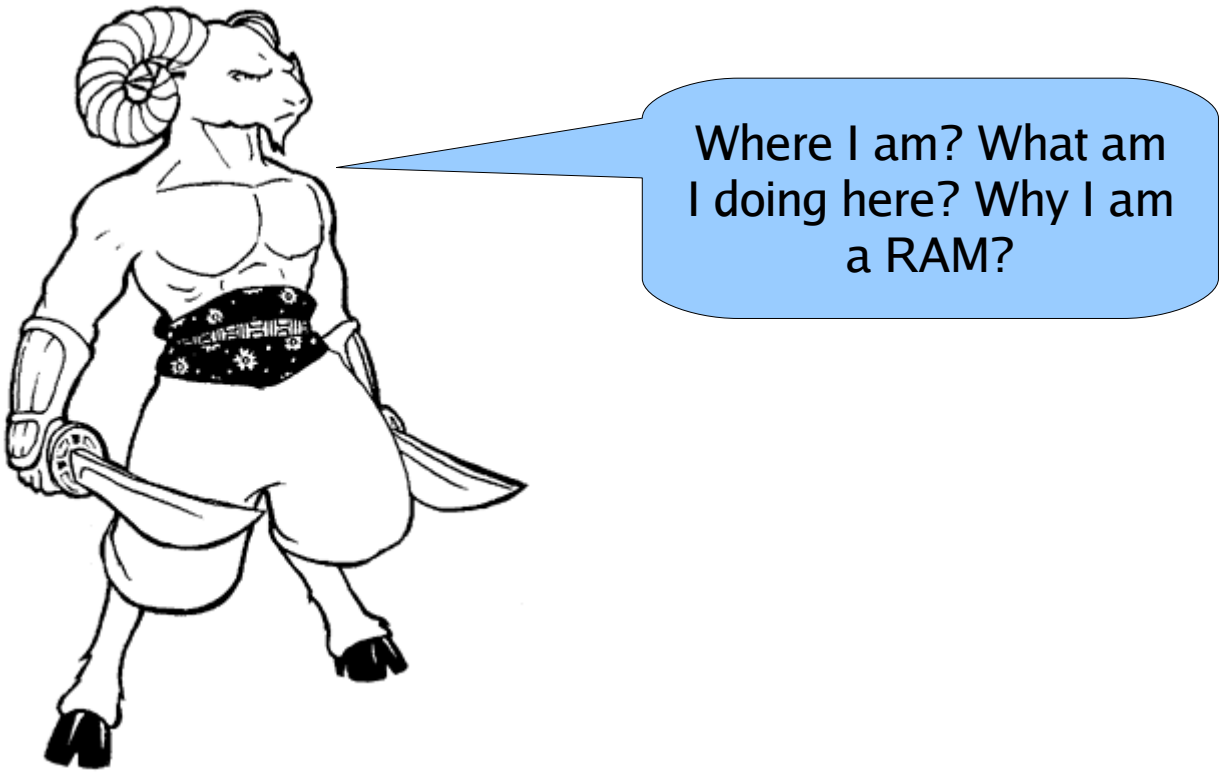
Reminder

- Exam #2 on Tuesday next week

I will study hard
in this exam.

RAM

- RAM is volatile
  - It forgets everything when the power is turned off

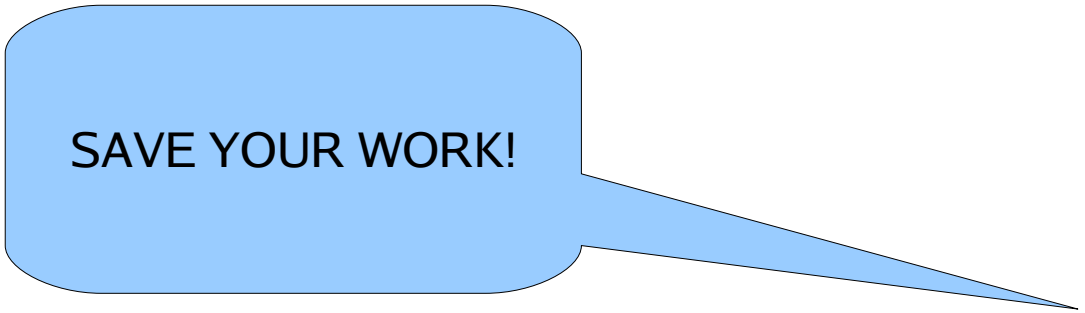


RAM

- Example: Casio FX-150
  - Battery powered pocket computer
  - 1680 Bytes of RAM
  - Can store 10 programs even when turned off
    - Electricity in memory
  - But when the battery dies... bye, bye programs.

RAM

- RAM volatility is one reason that the magnificent, infallible computer is vulnerable to the vagaries of outmoded, erratic power generating stations!
- That's why, only at UPLB computer labs will you hear the words:



ROM

- Once the contents are entered, can never be rewritten

Except for EPROM, Erasable Programmable ROM

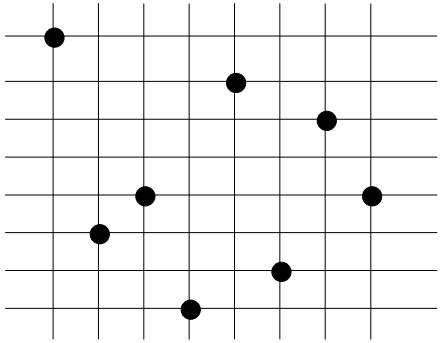
  - Ordinarily, ROM are programmed at the factory
  - But there are now also PROMs (Programmable ROMs) which can be custom-programmed to the user's specs.

What are you doing on Prom night?

Going to ICS lab to get fixed!

ROM

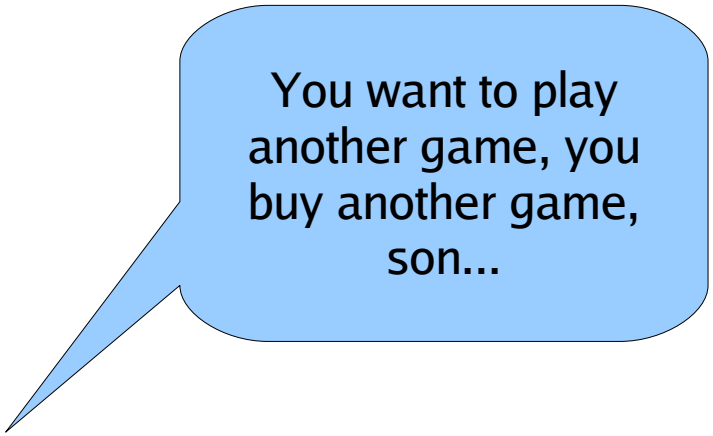
- Unlike RAM, ROM is non-volatile
  - it keeps its contents even without power
  - after all, its nothing but a huge grid of wires with physical connections at some intersections
  - the connections remain, regardless of electrical current



When I say “huge,” I mean “tiny”!

ROM

- Some typical uses of ROM:
  - Most video game cartidges are programmed in ROM
  - Just plug it in and it's ready to go
  - Of course it can't be reprogrammed



ROM

- Some typical uses of ROM:
  - Many personal computers have thousands of bytes of ROM
  - ROM store the program which allows tha machine to understand the language called BASIC

Hello, I'm ROM. I'm a Ferengi of Star Trek: Deep Space Nine. Your professor cannot find a picture of ROM so he used me instead.

PROGRAM

- Whether the memory is RAM or ROM, what really controls the computer is the PROGRAM

If programs really rule the computer, why is there no proper scientific name for it

Yeah, just like the dinosaurs of the past which rule the world.

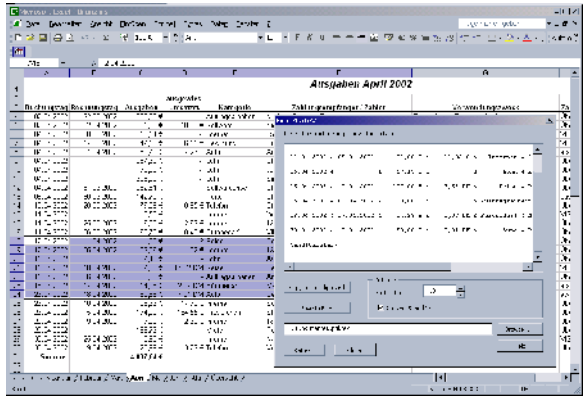
Lets call “program” something in latin or greek

What about technicalculus? Regula rationocerous? Cephaloneuralgia?

SOFTWARE

- But that's not how it is in Computer Science
- Instead, programs in general are called SOFTWARE
- To distinguish them from the circuit boards, cathode ray monitors, disk drives, keyboards and other items of computer HARDWARE

HARDWARE



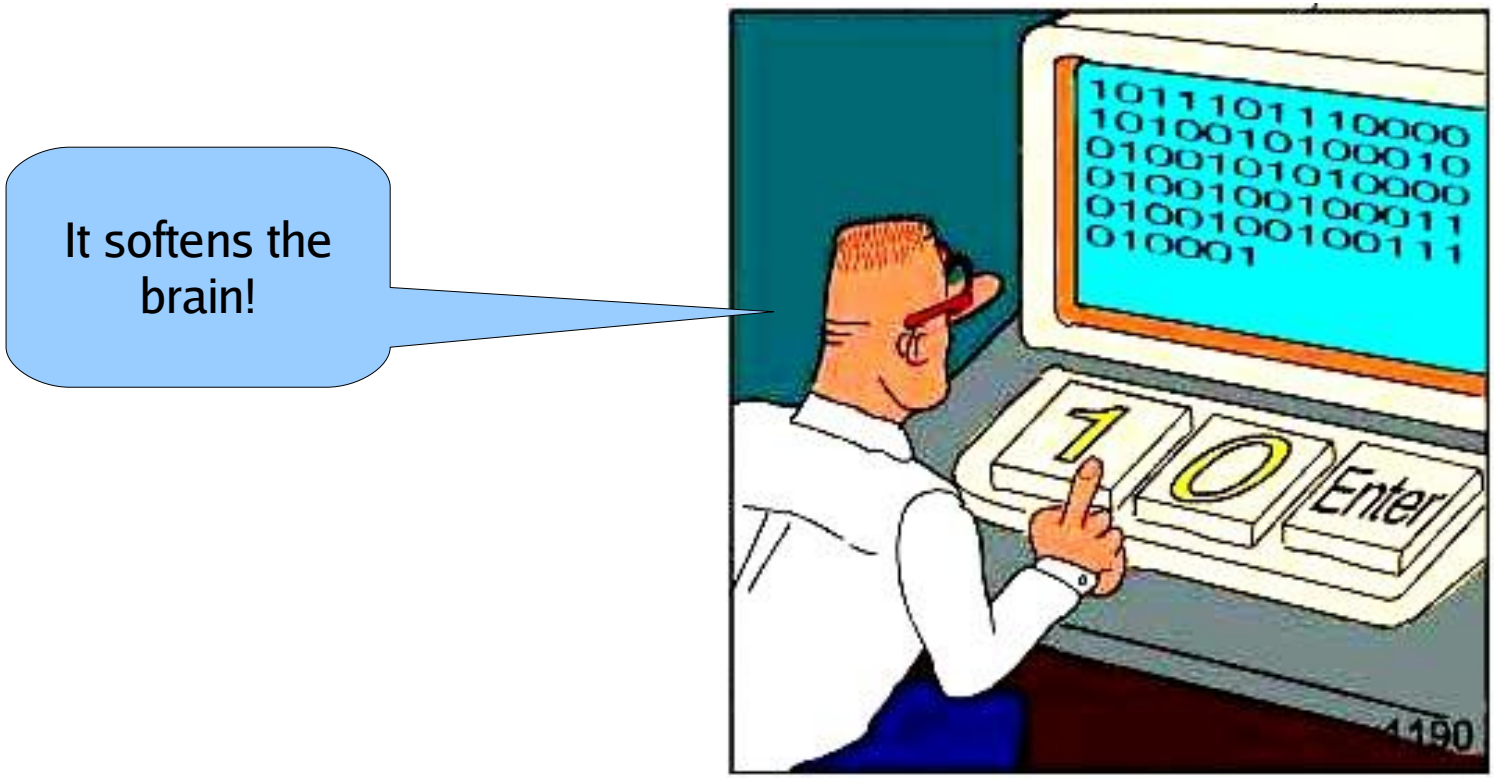
SOFTWARE

Tupperware

PEOPLEWARE

SOFTWARE

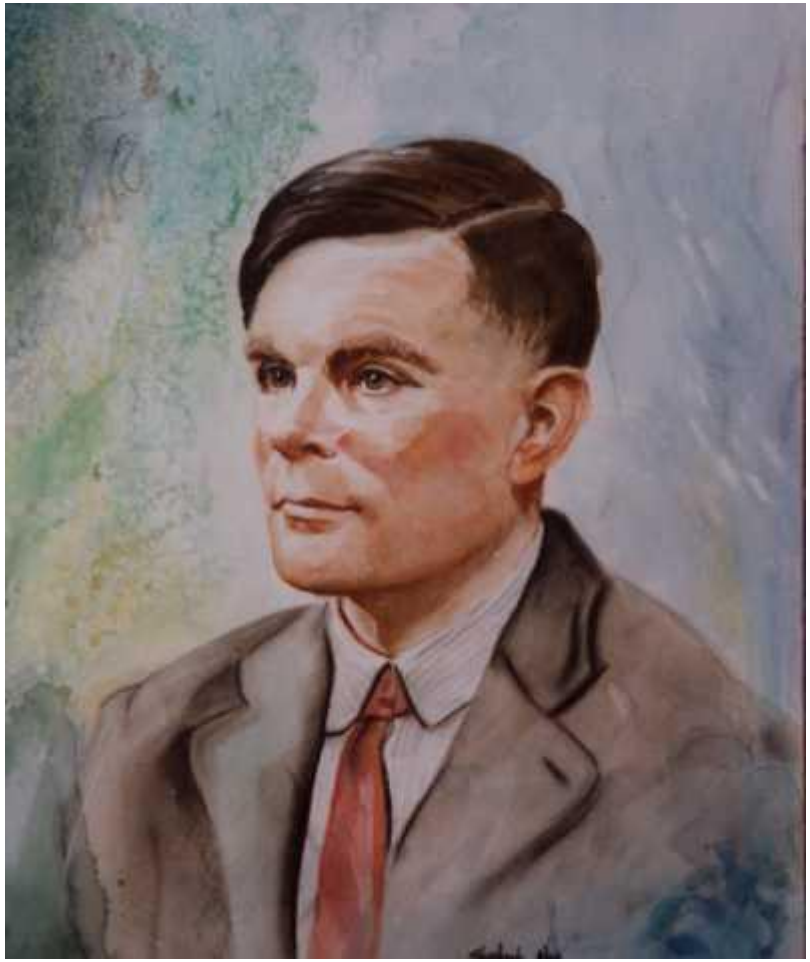
- What is really funny about the term SOFTWARE is that it is one of the HARDEST things about computing



REAL Programmers code in BINARY.

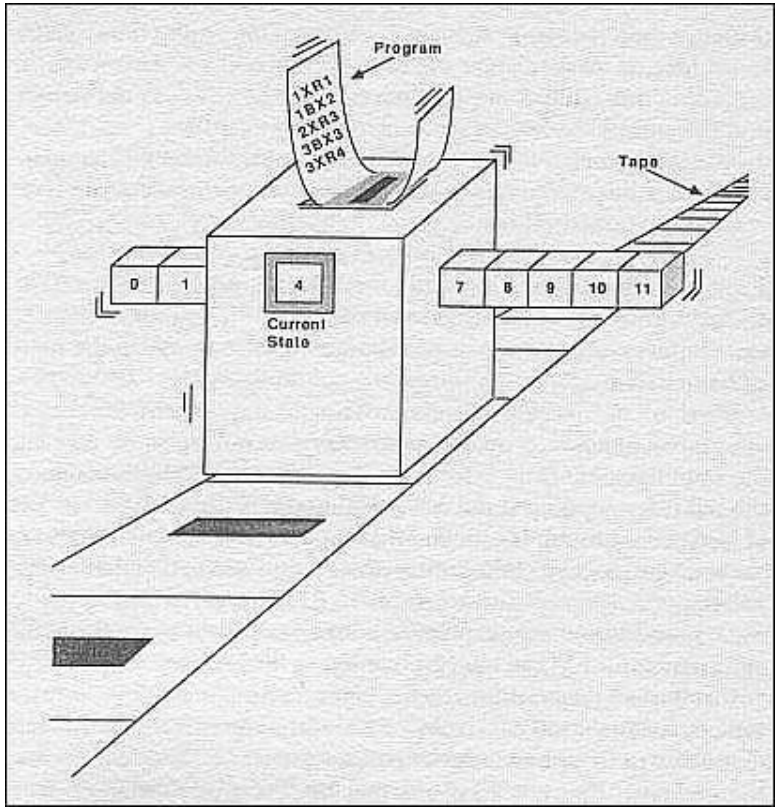
Alan Turing

- While Ada Lovelace was the original programmer, the first person to prove the full power of software was ALAN TURING (1912-1954)
- He dreamed up the TURING MACHINE
- Not real machines but abstract machine, existing only in theory



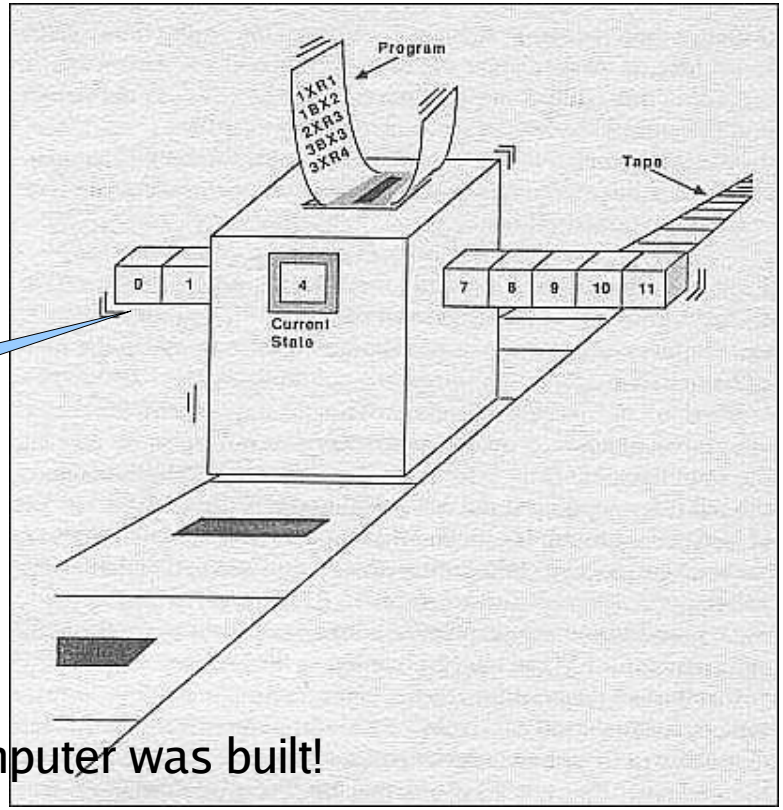
Turing Machine

- An I/O device similar to a black box
- Reads a sequence of 0s and 1s
- The output depends on the present input and the previous output
- The changes from one output state to the next are given by definite rules called TRANSITION RULES



Turing Machine

- A way of thinking physically about logic
- Any well-defined, step-by-step logical procedure can be embodied in some Turing machine



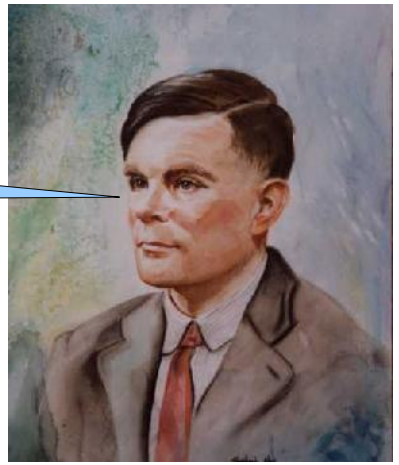
There's a Turning machine that can add!

Note: Turing thought of this 10 years before a real computer was built!

Turing Machine

- What Turning proved is that:

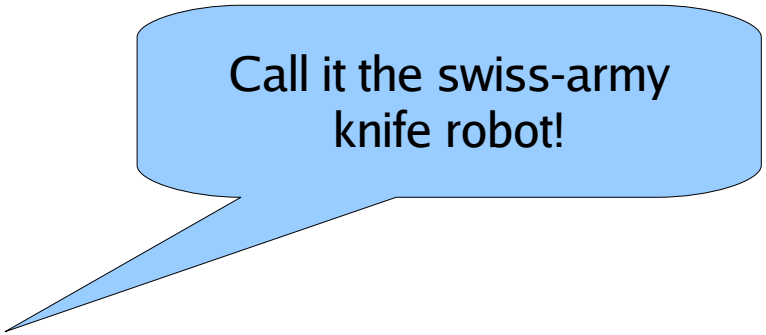
It is theoretically possible to construct a single Turing machine that can imitate all other Turing machines



- The UNIVERSAL TURING MACHINE
- The implication is staggering:
  - a single programmable machine can perform ANY well-defined step-by-step logical procedure

Turing Machine

- John von Neumann carried Turing's ideas a step further.
- von Neumann realized that one could:
  - build a machine X which builds other machines from plans encoded as input to X
  - Self-reproducing Machines are possible

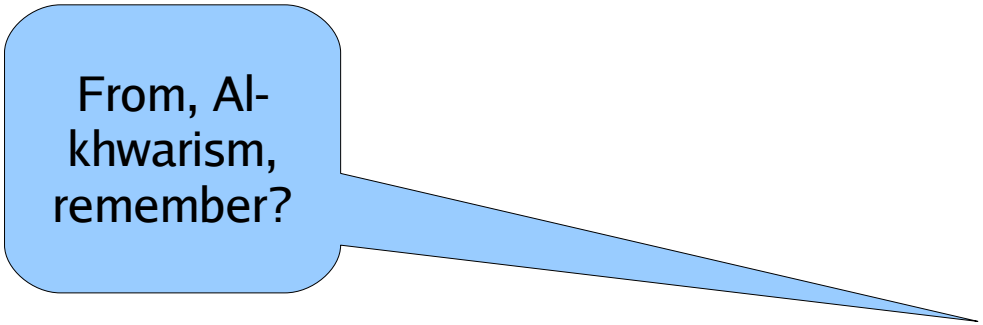


The Computer

- The Digital Computer is a fancy Universal Turing Machine come to life!
- Therefore, as Turing proved:
  - it can do ANYTHING (or, more accurately, SIMULATE anything)
  - the only limit is the amount of time at the user's disposal
  - say from now until the death of the solar system

The Computer

- To be perfectly honest, there are a couple of other qualifications on the “ANYTHING”
- What kind of of “ANYTHING” can a computer do?
- IOW, computers do ALGORITHMS



Algorithm

- An algorithm is simply an **well defined** **step-by-step** procedure:
  - **step-by-step** means each step is completed before the next is begun
    - just like a stair

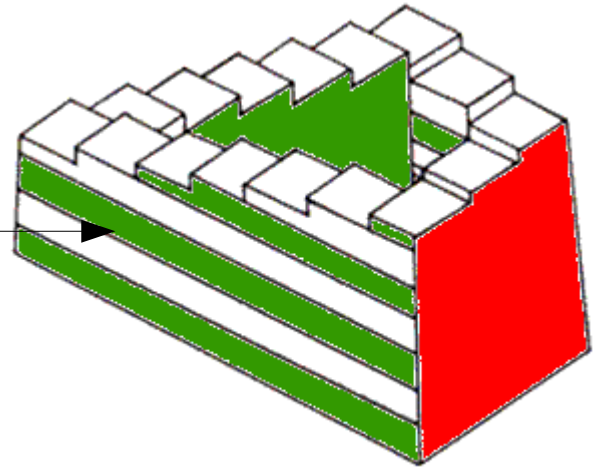
You need to climb each step before you can reach the top

Unless, of course, if you are a giant.

Algorithm

- An algorithm is simply an **well defined** **step-by-step** procedure:
  - **well defined**, meaning each step is completely determined by current input and the results of previous steps.
    - No ambiguity is allowed!

Ambiguious staircase?



Algorithm

- Examples: Is this an algorithm?

If nuclear warheads are falling like raindrops during a typhoon, I will lie down and try to enjoy it. Otherwise, I will go to class as usual.

Algorithm

- yes it is!
- Why?

If nuclear warheads are falling like raindrops during a typhoon, I will lie down and try to enjoy it. Otherwise, I will go to class as usual.

- Because you always know what to do:
  - check to see if warheads are falling
  - if yes, lie down and enjoy
  - if no, go to class

Algorithm

- Likewise, algebraic formulas represent algorithms
- Example: $y = x^2 + 2x + 10$ means:

  (1) input a number x

  (2) multiply x times itself

  (3) multiple x times 2

  (4) add the result of (2) and (3)

  (5) add 10 to the result of (4)

If you understand this, lie down and enjoy yourself. If not, go to your MATH 11 class!

Algorithm

- Example of non-algorithm:

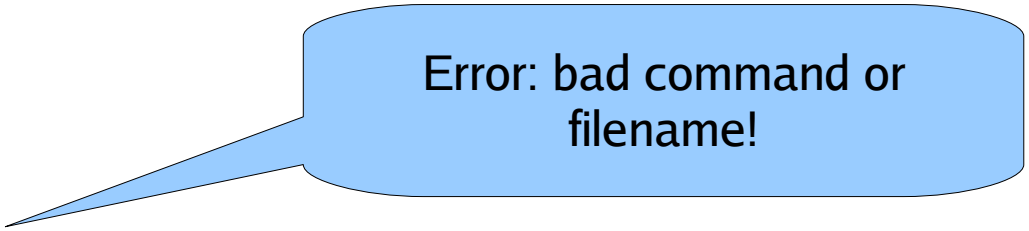
If nuclear warheads are falling like raindrops during a typhoon, I will lie down and try to enjoy it.

And in the meantime, what do you do?

- This fails to tell you what to do if no warheads are falling... so it is not **well defined**!

Algorithm

- Another non-algorithm:
  $y = x^2 + @\ 2x - 10$
- This is no algorithm because it is not expressed in proper algebraic grammar
- We do not have a meaning to the symbols + @
- If you try to make a computer do a non-algorithm, it will just sit there flashing error messages.



Last Reminder

- Exam #2 on Tuesday next week
- CMSC 11 holiday on Thursday next week: Palarong UPLB

Ano bang chuva ek-ek ito. Convert hex into logic diagram using only NAND gates. Naalala ko tuloy si Nando na president ng frat sa kabilang tambayan. Ang cute niya, in fairness. Sana makipag-rambulan ang sorority, este, frat namin sa kanila. Magpapahuli talaga ako sa kanya...