Telling a computer how to behave

(via pseudocode -- a workaround for Computing’s Tower of Babel.)

COS 116, Spring 2011
Sanjeev Arora

The New York Times

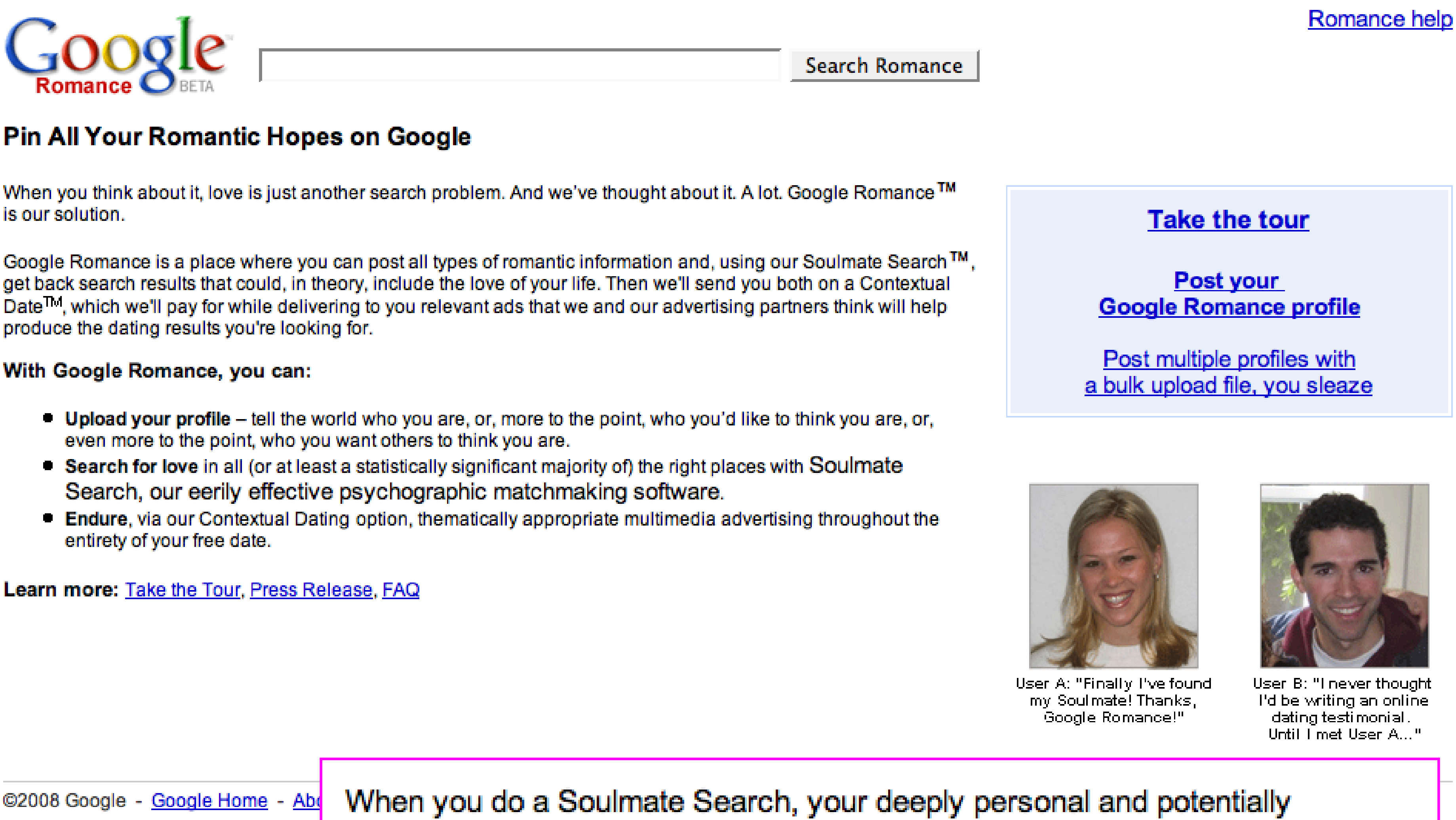
FINDINGS

Hitting It Off, Thanks to Algorithms of Love

Viktor Koen

By JOHN TIERNEY

Jan 29, 2008



Pin All Your Romantic Hopes on Google

When you think about it, love is just another search problem. And we've thought about it. A lot. Google Romance™ is our solution.

Google Romance is a place where you can post all types of romantic information and, using our Soulmate Search™, get back search results that could, in theory, include the love of your life. Then we'll send you both on a Contextual Date™, which we'll pay for while delivering to you relevant ads that we and our advertising partners think will help produce the dating results you're looking for.

With Google Romance, you can:

- **Upload your profile** – tell the world who you are, or, more to the point, who you'd like to think you are, or, even more to the point, who you want others to think you are.
- **Search for love** in all (or at least a statistically significant majority of) the right places with Soulmate Search, our eerily effective psychographic matchmaking software.
- **Endure**, via our Contextual Dating option, thematically appropriate multimedia advertising throughout the entirety of your free date.

Learn more: Take the Tour, Press Release, FAQ

Take the tour

Post your Google Romance profile

Post multiple profiles with a bulk upload file, you sleaze

User A: "Finally I've found my Soulmate! Thanks, Google Romance!"

User B: "I never thought I'd be writing an online dating testimonial. Until I met User A..."

©2008 Google - Google Home - Abo

When you do a Soulmate Search, your deeply personal and potentially life-altering search results are produced solely by computer algorithm, without human intervention of any kind.

Note: depending on your personality, you may or may not find this reassuring.

Discussion
Time

Example: Adding two numbers

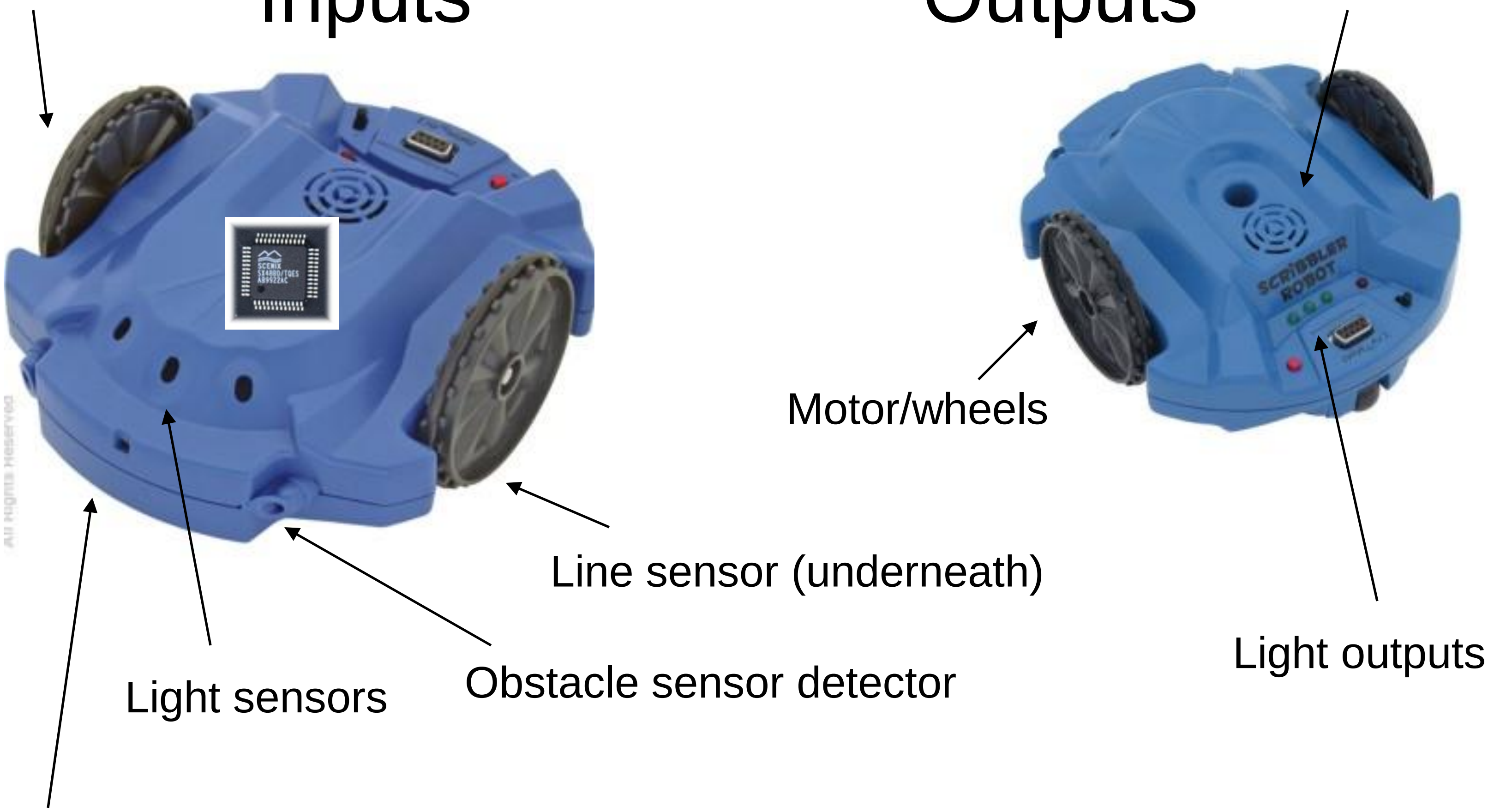
Imagine you are describing this task to somebody who has never done it. How would you describe it?

How can we describe an algorithm precisely enough so there is no ambiguity?

Last time: Scribbler

Inputs

Outputs



Stall sensor

Speaker

Motor/wheels

Line sensor (underneath)

Light outputs

Light sensors

Obstacle sensor detector

Obstacle sensor emitter

Scribbler’s “Language”

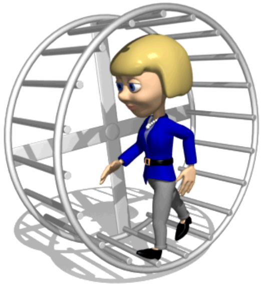
- Several types of simple instructions
  - E.g. “Move forward for 1 s”
- Two types of compound instructions

Conditional

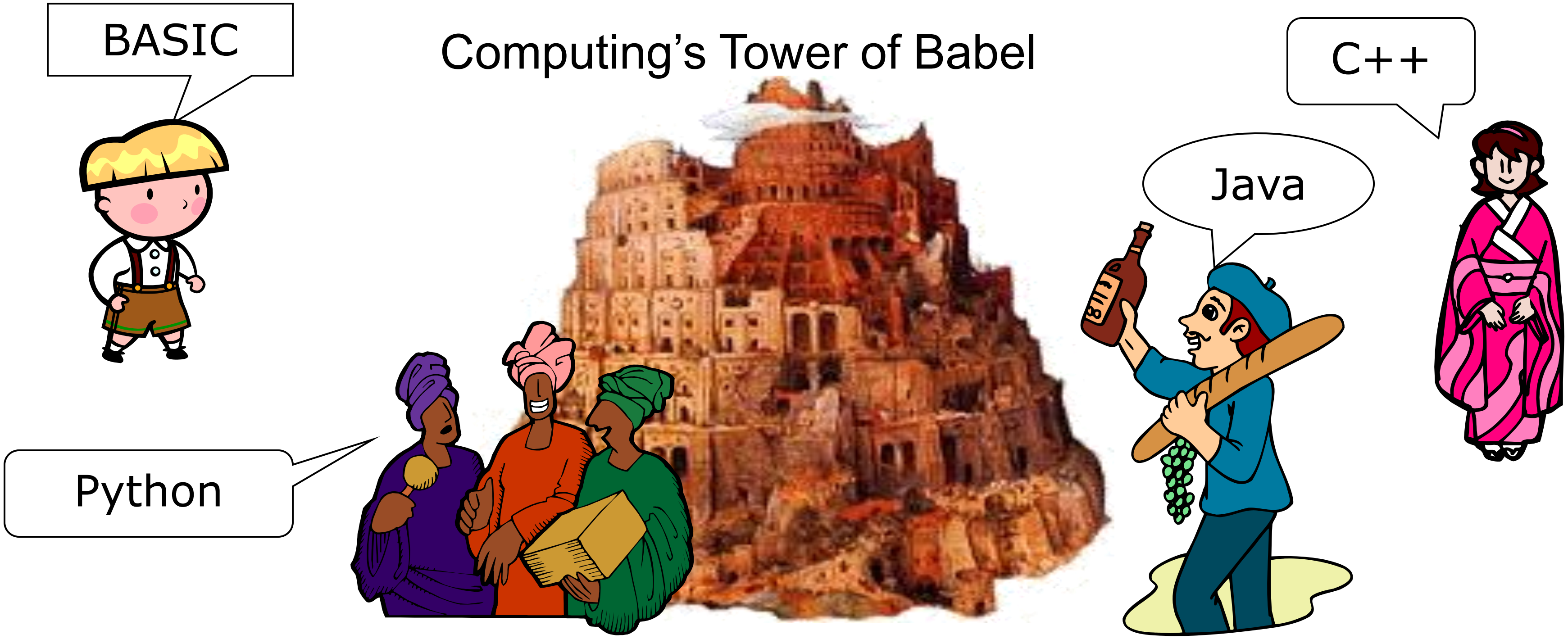
```
If condition Then
{
        List of instructions
}
Else
{
        List of instructions
}
```

Loop

```
Do 5 times
{
      List of instructions
}
```



Scribbler language illustrates essential features of all computer languages

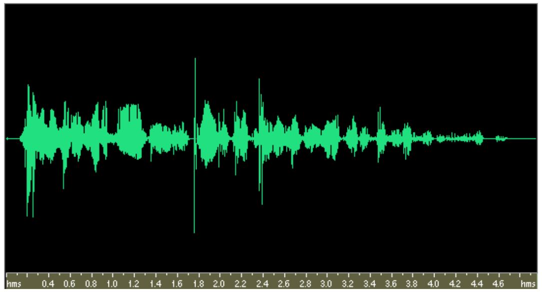

Computing’s Tower of Babel

Features of human languages: nouns/verbs/subjects/objects, etc.

Features of computer languages: variables, simple arithmetic instructions, conditional/loop statements

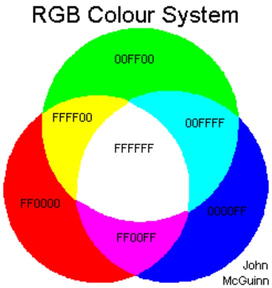
For a computer, everything’s a number

Justin Bieber song



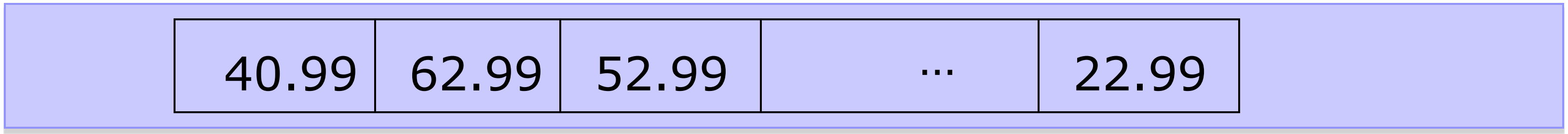
Sequence of Numbers representing frequency, amplitude, etc.at every instant (e.g., mp3 format)

Image



Sequence of Numbers representing color value of each pixel.

General task faced by computer



40.99	62.99	52.99	…	22.99

Given: Sequence of numbers in memory (eg, mp3 song)

Goal: Process these numbers to achieve a desired sequence of numbers as output (eg, electrical signals that drive the headphones and reproduce the song)

A simple problem

- Our robot is getting ready for a big date…

- How would it identify the cheapest bottle? (Say it can scan prices)

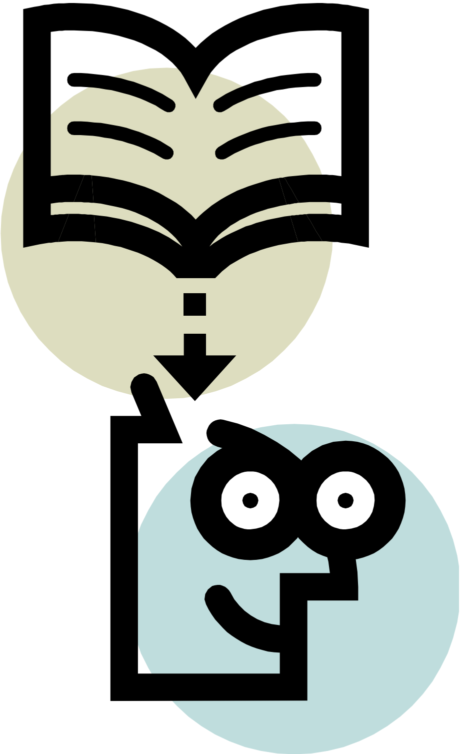
Solution

- Pick up first bottle, check price

- Walk down aisle. For each bottle, do this:
  - If price on bottle is less than price in hand, exchange it with the one in hand.

Similar question in different setting

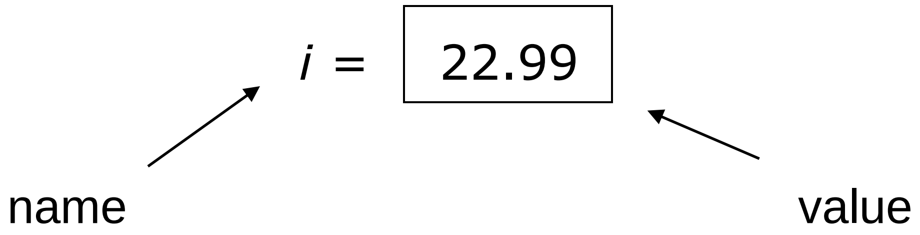
- Robot has *n* prices stored in memory
- Want to find minimum price



Memory: a simplified view

- A scratchpad that can be perfectly erased and re-written any number of times

- A variable: a piece of memory with a name; stores a "value"



i = 22.99

name

value

Examples

$i \leftarrow 5$	Sets *i* to value 5
$j \leftarrow i$	Sets *j* to whatever value is in *i*. Leaves *i* unchanged
$i \leftarrow j + 1$	Sets *i* to *j* + 1. Leaves *j* unchanged
$i \leftarrow i + 1$	Sets *i* to 1 more than it was.

Arrays

- *A* is an array of *n* values
  A[*i*] is the *i*'th value

A =	40.99	62.99	52.99	…	22.99

- Example: *A*[3] = 52.99

Solution

- Pick up first bottle, check price
- Walk down aisle. For each bottle, do this:
  - If price on bottle is less than price in hand, exchange it with the one in hand.

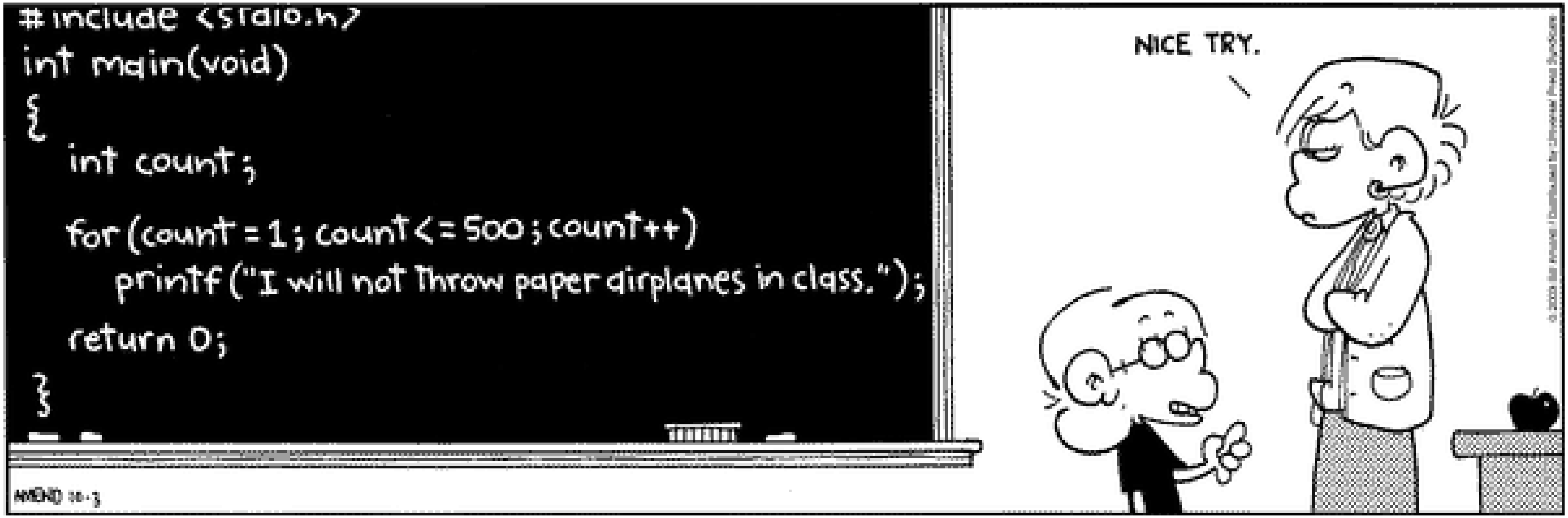
Procedure findmin

- *n* items, stored in array *A*
- Variables are *i*, *best*

```
$best \leftarrow 1$
Do for $i = 2$ to $n$
   {
         if ( $A[i] < A[best]$ ) then
               $best \leftarrow i$
   }
```

Another way to do the same

```
$best \leftarrow 1$;
$i \leftarrow 1$
Do while ($i$ < n)
{
        $i \leftarrow i + 1$;
        if ( $A[i] < A[best]$ ) then
                $best \leftarrow i$
}
```



```
#include <stdio.h>
int main(void)
{
   int count;

   for(count=1;count<=500;count++)
      printf("I will not Throw paper airplanes in class.");
   return 0;
}
```

New problem for robot: sorting

Arrange them so prices increase from left to right.

I have to sort n bottles.
Let me find the cheapest bottle and move it to leftmost position. Then I only have to sort n-1 bottles to its right.

Solution

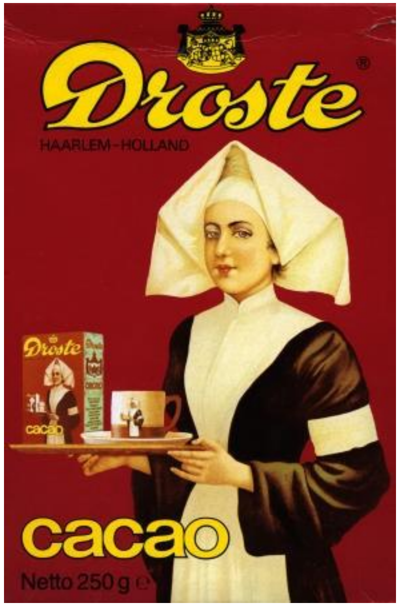
Note: we know how to do this!

```
Do for $i$=1 to $n$-1
{
   Find cheapest bottle among those numbered $i$ to $n$

   Swap that bottle and the $i$’th bottle.
}
```

“selection sort”

Recursion: When a computational procedure is defined in terms of itself.



Swapping

- Suppose x and y are variables.
  How do you swap their values?

- Need extra variable!

```
$tmp \leftarrow x$
$x \leftarrow y$
$y \leftarrow tmp$
```

Algorithm

- A precise unambiguous procedure for accomplishing a computational task

- Named for Abu Abdullah Muhammad bin Musa al-Khwarizmi
  - His book "Al-Jabr wa-al-Muqabilah" evolved into today's high school algebra text.

- Examples:
  recipe, long division, selection sort.

Love, Marriage, and Lying

Standard disclaimer.

Stable Matching Problem

Problem:

Given *N* men & *N* women, find "suitable" matching

- Everyone lists their preferences from best to worst.

Men's Preference List

Man	1st	2nd	3rd	4th	5th
Victor	Bertha	Amy	Diane	Erika	Clare
Wyatt	Diane	Bertha	Amy	Clare	Erika
Xavier	Bertha	Erika	Clare	Diane	Amy
Yancey	Amy	Diane	Clare	Bertha	Erika
Zeus	Bertha	Diane	Amy	Erika	Clare

↑ best

↑ worst

Stable Matching Problem

Problem:

Given *N* men & *N* women, find "suitable" matching

- □ Everyone lists their preferences from best to worst.

Women's Preference List

Woman	1st	2nd	3rd	4th	5th
Amy	Zeus	Victor	Wyatt	Yancey	Xavier
Bertha	Xavier	Wyatt	Yancey	Victor	Zeus
Clare	Wyatt	Xavier	Yancey	Zeus	Victor
Diane	Victor	Zeus	Yancey	Xavier	Wyatt
Erika	Yancey	Wyatt	Zeus	Xavier	Victor

best

worst

HW (due in class on Thurs)

- Propose (in a para or two) an algorithm for the stable matching problem. Be clear and brief.

  As part of your solution, formulate what a “good solution” means and indicate whether your algorithm achieves it.

(counts towards your participation grade)