Infinite Loops

- Which of these subroutines terminate for all initial values of $n$?

```python
def ex1(n):
    while n < 14:
        print n
        n = n + 1

def ex2(n):
    while True:
        print n
        n = n - 1

def ex3(n):
    while n < 21:
        print n
        n = n - 1

def ex4(n):
    while n > 100:
        print n
        n = n + 1

def ex5(n):
    while n > 0:
        print n
        n = n - 1

def ex6(n):
    while False:
        print n
        n = n + 1
```
While True

- What sorts of program would *purposely* have an infinite loop?
- Think about a software-controller thermostat. It might have a program that looks something like this.

```python
def thermostat(low, high):
    while True:
        t = currentTemp()
        if t >= high:
            runAC(4)
        elif t < low:
            runHeat(2)

thermostat(68, 75)
```

Loop Forever

- operating systems
- user interfaces
- video games
- process controllers
- robots
Robot Basics

- From a software standpoint, modern robots are just computers.
- Typically they have less memory and processing power than a standard computer.
- Sensors and effectors under software control.

Standard Robots

- Industrial manufacturing robots.
- Research /hobby robots.
- Demonstration robots.
- Home robots.
- Planetary rovers.
- Movie robots.
Manufacturing

- Often arms, little sending.
- Painting. Part sorting.
- Painting.
- Repeatable actions.

Research / Hobby

- Pioneer
- Handy Board / Lego
- Segbot
- Stanley
Space Exploration
- Sojourner
- Deep Space Agent

Home Robots
- Roomba.
- Mowers.
- Moppers.
- Big in Japan.
- Elderly aids.
- Emergency rescue bots, Aibo.
Demonstration Robots

- Honda: Asimo.
- Toyota’s robot.
- Sony: QRio.

Sensors and Effectors

- Sensors:
  - bump
  - infrared
  - vision
  - light
  - sonar
  - sound

- Effectors:
  - motors
  - lights
  - sounds
  - graphical display
  - laser
Simple Learning

- Words: “hello”, “don’t do that”, “sit”, “stand up”, “lie down”

Example Code

```python
act[0] = 0
act[1] = 0
actions = ["lay6", "sit2", "sit4", "stand2", "stand9"]
lastact = 0
while True:
    cmd = Voice()
    if cmd == "sit":
        doAction(actions[act[0]])
        lastact = 0
    elif cmd == "stand":
        doAction(actions[act[1]])
        lastact = 1
    elif cmd == "good Aibo":
        doAction("happy")
    elif cmd == "bad dog":
        doAction("sad sound")
    act[lastact] = (act[lastact] + 1) % 4
```
Trainer: In Words

- For each recognized voice command, there is an associated action program.
- When a voice command is recognized, the corresponding action is taken.
- On hearing “Good Aibo”, nothing needs to change.
- On “Bad dog”, we know the most recent command needs a different action program. It is incremented to the next on the list.

Scribbler

- Demonstrate sensors (bump, light, stall, line).
- Demonstrate effectors (light, sound/music, motors).
- Simple loops, etc.
**Simple Program**

- lights(cmd): reports the value (0-255) of the light sensors. cmd can be “avg” (average value), “max”, “min”, “lxx” (left sensor), “nln” (center minus sides), etc.
- Jumps forward if lights on.
- move(l,r,d): power to left and right motors (-100 to 100) and duration (0 means continue until next motor command).

```
while True:
    if lights("avg") > 10:
        move(50,50,0)
    else:
        move(0,0,0)
```

**Other Commands**

- led(cmd): Turn on LEDs. For example, cmd of “lxn” means turn left light on and pright light off, leaving middle light in its current state.
- play(cmd): Plays give sound command on speaker.
- stall(): Is robot stalled?
- wall(loc): loc of “wx” means sensing wall on left, not right.
- line(loc): loc of “xl” means line on right, not left.
- Programs compiled into “stamp BASIC”
Questions

• Match robot program to the resulting behavior.

Next Time

• Simulation and Modeling
• Dinesh Pai visits.