L14: Discussion

And practice questions

Lucas A. J. Bastien

E7 Spring 2017, University of California at Berkeley

February 17, 2017

Version: release

Announcements

Lab 05 is due on February 24 at 12 pm (noon)

Today:

- Discussion, Practice questions
- Written feedback

Next Week:

- Monday: holiday
- Wednesday: Data visualization and plotting
- Time complexity of algorithms

Grades, solutions, re-grade requests, and partial credit

Grades for lab 01 and lab 02 should be posted later today or this weekend

Solutions will be posted on bCourses (in "Files") as PDF files

Re-grade requests:

- **No later than one week after grades are released**, rounded up to 11:59pm of the last day
- Guidelines are written up as a bCourses Page (read carefully)
- Re-grade requests that do not follow these guidelines will not be considered

Grades, solutions, re-grade requests, and partial credit

Grades for lab 01 and lab 02 should be posted later today or this weekend

Solutions will be posted on bCourses (in "Files") as PDF files

Re-grade requests:

- **No later than one week after grades are released**, rounded up to 11:59pm of the last day
- Guidelines are written up as a bCourses Page (read carefully)
- Re-grade requests that do not follow these guidelines will not be considered

Partial credit:

- If you cannot make your function work for **all** test cases, make your function work for **some** test cases to get partial credit

Code snippets from Python's Numpy package

Numpy is a package to create and manipulate arrays for scientific computing (and other applications) in Python

```
def matrix_power(M, n):
    """
    Raise a square matrix to the (integer) power `n`.
```

...

```
    result = M
    if n <= 3:
        for _ in range(n-1):
            result=N.dot(result, M)
        return result

    # binary decomposition to reduce the number of Matrix
    # multiplications for n > 3.
    beta = binary_repr(n)
    Z, q, t = M, 0, len(beta)
    while beta[t-q-1] == '0':
        Z = N.dot(Z, Z)
        q += 1
    result = Z
    for k in range(q+1, t):
        Z = N.dot(Z, Z)
        if beta[t-k-1] == '1':
            result = N.dot(result, Z)
    return result
```

Code snippets from numpy/matrixlib/defmatrix.py (http://www.numpy.org/), license: BSD

Code snippet from the CMAQ model

CMAQ is the Community Multi-scale Air Quality Model, written in Fortran. It is used to research and forecast air pollution problems

```
C Do the gridded computation for horizontal diffusion

C Get the contravariant eddy diffusivities

      CALL HCDIFF3D ( FDATE, FTIME, K11BAR3D, K22BAR3D, DT )

C get number of steps based on eddy time

      NSTEPS = INT ( DTSEC / DT ) + 1
      DT = DTSEC / FLOAT( NSTEPS )

      WRITE( LOGDEV,1005 ) DT, NSTEPS

      DTDX1S = DT * RDX1S
      DTDX2S = DT * RDX2S

      DO L = 1, NLAYS
         DO R = STARTROW, ENDROW          !  DO R = 1, NROWS + 1
            DO C = STARTCOL, ENDCOL       !     DO C = 1, NCOLS + 1
               RK11( C,R,L ) = RK11( C,R,L ) * K11BAR3D( C,R,L )
               RK22( C,R,L ) = RK22( C,R,L ) * K22BAR3D( C,R,L )
            END DO
         END DO
      END DO
```

Code snippet retrieved from https://github.com/USEPA/CMAQ/blob/5.1/models/CCTM/hdiff/multiscale/hdiff.F (commit: 6e56fcf02862b1c676c9144e215a0125fa1cd9d2, retrieved February 17$^{\text{th}}$ 2017). License: see complete file

More on the IEEE-754 floating point representations

Which of the following quantities can be represented using the double precision representation of the IEEE-754 standard?

(A) 0.1

(B) 0.15

(C) 0.2

(D) 0.3

(E) 0.75

(F) 1

(G) 3

More on the IEEE-754 floating point representations

Which of the following quantities can be represented using the double precision representation of the IEEE-754 standard?

(A) 0.1

(B) 0.15

(C) 0.2

(D) 0.3

(E) 0.75 $\leftarrow 0.75 = 2^{-1} + 2^{-2}$

(F) 1 $\leftarrow 1 = 2^{0}$

(G) 3 $\leftarrow 3 = 2^{0} + 2^{1}$

Only numbers that can be written as $2^i + 2^j + 2^k + \ldots$ (where $i, j, k, \ldots$ are integers) can be represented using the floating point representations (single or double precision) of the IEEE-754 standard

More on the IEEE-754 floating point representations

```
>> fprintf('%.40f\n', 0.1)
0.1000000000000000055511151231257827021182

>> fprintf('%.40f\n', 0.15)
0.1499999999999999944888487687421729788 18

>> fprintf('%.40f\n', 0.2)
0.2000000000000000111022302462515654042363

>> fprintf('%.40f\n', 0.3)
0.2999999999999999888977697537484345957637

>> fprintf('%.40f\n', 0.75)
0.7500000000000000000000000000000000000000

>> fprintf('%.40f\n', 1)
1.0000000000000000000000000000000000000000

>> fprintf('%.40f\n', 3)
3.0000000000000000000000000000000000000000
```

More on the IEEE-754 floating point representations

Which of the following logical expressions evaluate to true?

(A) `0.3 == 0.3`

(B) `0.1+0.2 == 0.3`

(C) `0.5+0.75 == 1.25`

(D) `1e6 == 1e6`

(E) `1e6+1 == 1e6`

(F) `1e20 == 1e20`

(G) `1e20+1 == 1e20`

More on the IEEE-754 floating point representations

Which of the following logical expressions evaluate to true?

(A) `0.3 == 0.3`

(B) `0.1+0.2 == 0.3`

(C) `0.5+0.75 == 1.25`

(D) `1e6 == 1e6`

(E) `1e6+1 == 1e6`

(F) `1e20 == 1e20`

(G) `1e20+1 == 1e20`

More on the IEEE-754 floating point representations

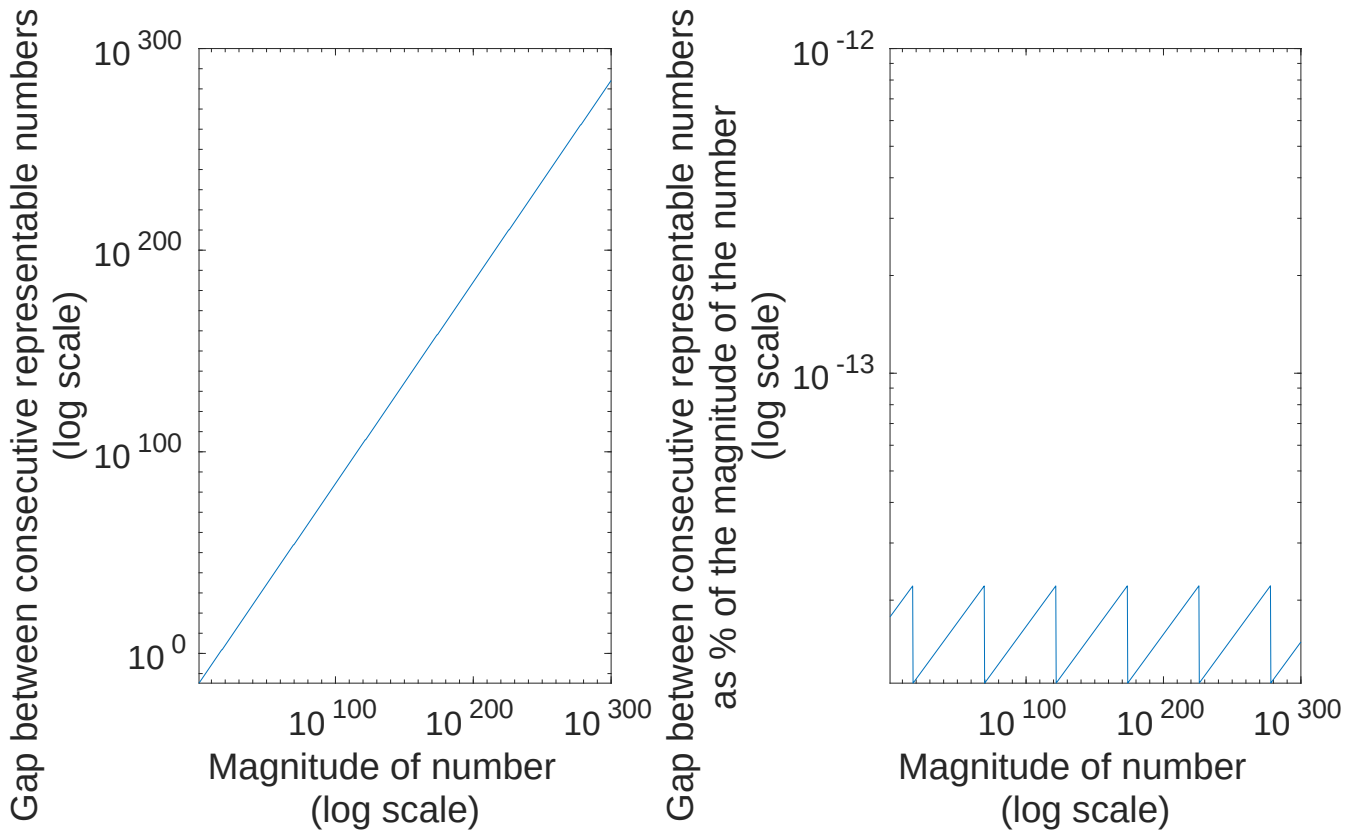
```
>> fprintf('%.40f\n%.40f\n', 0.1+0.2, 0.3)
0.3000000000000000444089209850062616169453
0.2999999999999999888977697537484345957637

>> % The gap between consecutive representable numbers
>> % around 1e6 is a lot smaller than 1
>> eps(1e6)
ans =
   1.1642e-10
>> 1e6+1 == 1e6
ans =
  logical
   0

>> % The gap between consecutive representable numbers
>> % around 1e20 is a lot bigger than 1
>> eps(1e20)
ans =
       16384
>> 1e20+1 == 1e20
ans =
  logical
   1
```

More on the IEEE-754 floating point representations

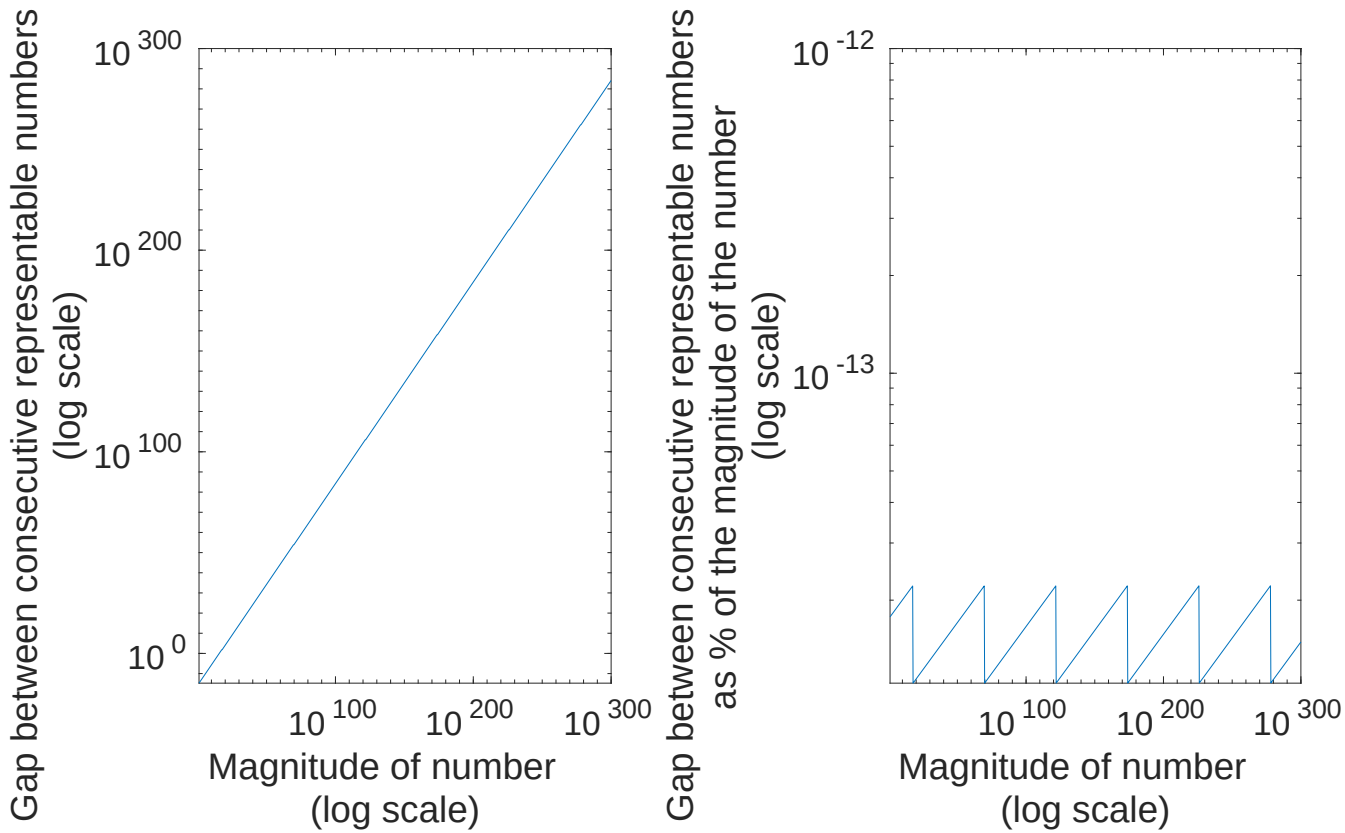
- **Left panel:** The accuracy of the representation becomes lower as numbers get bigger



The figure is for the double precision representation

More on the IEEE-754 floating point representations

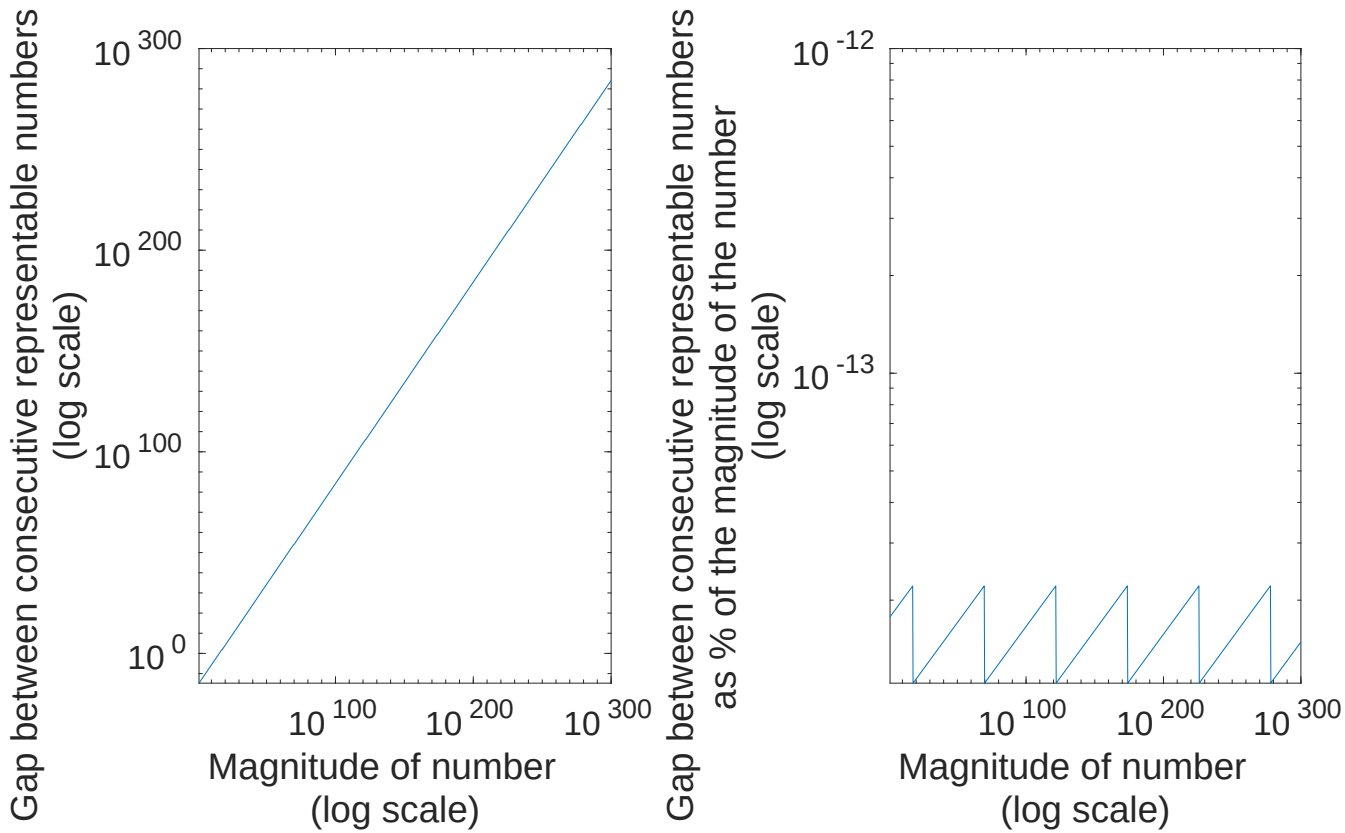
- **Left panel:** The accuracy of the representation becomes lower as numbers get bigger
- **Right panel:** The relative accuracy of the representation is high no matter how big the number is



The figure is for the double precision representation

More on the IEEE-754 floating point representations

- **Left panel:** The accuracy of the representation becomes lower as numbers get bigger
- **Right panel:** The relative accuracy of the representation is high no matter how big the number is
- In the "basic" binary representation of numbers (*e.g.*, unsigned integers), numbers have constant spacing (1 for unsigned integers)



The figure is for the double precision representation

Appending values to vectors

In class exercise:

We define the variable v as the following vector:

```
>> v = [2, 9, -1, 3, 5];
```

Propose three ways to append the value 7 at the end of this vector (and store the result in the variable v)

Appending values to vectors

In class exercise:

We define the variable v as the following vector:

```
>> v = [2, 9, -1, 3, 5];
```

Propose three ways to append the value 7 at the end of this vector (and store the result in the variable v)

---

1. Use indexing and the keyword `end` (or the function `numel`):

```
v(end+1) = 7;
v(numel(v)+1) = 7;
```

Appending values to vectors

In class exercise:

We define the variable v as the following vector:

```
>> v = [2, 9, -1, 3, 5];
```

Propose three ways to append the value 7 at the end of this vector (and store the result in the variable v)

---

1. Use indexing and the keyword `end` (or the function `numel`):

```
v(end+1) = 7;
v(numel(v)+1) = 7;
```

---

2. Use a comma and square brackets:

```
v = [v, 7];
```

Appending values to vectors

In class exercise:

We define the variable v as the following vector:

```
>> v = [2, 9, -1, 3, 5];
```

Propose three ways to append the value 7 at the end of this vector (and store the result in the variable v)

---

1. Use indexing and the keyword `end` (or the function `numel`):

```
v(end+1) = 7;
v(numel(v)+1) = 7;
```

---

2. Use a comma and square brackets:

```
v = [v, 7];
```

With a column vector, use a semi-colon instead: `v = [v; 7];`

Appending values to vectors

In class exercise:

We define the variable v as the following vector:

```
>> v = [2, 9, -1, 3, 5];
```

Propose three ways to append the value 7 at the end of this vector (and store the result in the variable v)

---

1. Use indexing and the keyword `end` (or the function `numel`):

```
v(end+1) = 7;
v(numel(v)+1) = 7;
```

---

2. Use a comma and square brackets:

```
v = [v, 7];
```

With a column vector, use a semi-colon instead: `v = [v; 7];`

---

3. Use the built-in function `horzcat`:

```
v = horzcat(v, 7);
```

Appending values to vectors

In class exercise:

We define the variable v as the following vector:

```
>> v = [2, 9, -1, 3, 5];
```

Propose three ways to append the value 7 at the end of this vector (and store the result in the variable v)

---

1. Use indexing and the keyword `end` (or the function `numel`):

```
v(end+1) = 7;
v(numel(v)+1) = 7;
```

---

2. Use a comma and square brackets:

```
v = [v, 7];
```

With a column vector, use a semi-colon instead: `v = [v; 7];`

---

3. Use the built-in function `horzcat`:

```
v = horzcat(v, 7);
```

With a column vector, use `vertcat` instead: `v = vertcat(v, 7);`

The continue command

The `continue` command is used to skip to the end of the current iteration of a `for` or `while` loop (only for the inner-most loop containing the `continue` command)

What will the value of the variable "s" be after executing the following code?

```
>> s = 0;
>> for i = 1:10
>>     if i >= 4 & i < 9
>>         continue
>>     end
>>     s = s + 1;
>> end
```

(A) 3

(B) 4

(C) 5

(D) 9

The continue command

The `continue` command is used to skip to the end of the current iteration of a `for` or `while` loop (only for the inner-most loop containing the `continue` command)

What will the value of the variable "s" be after executing the following code?

```
>> s = 0;
>> for i = 1:10
>>     if i >= 4 & i < 9
>>         continue
>>     end
>>     s = s + 1;
>> end
```

(A) 3

(B) 4

(C) 5

(D) 9

The first actual computer bug

- Below: a page of the Harvard Mark II computer log (late 1940s)
- A moth was found in the machine and taped inside the log
- The term "bug" (for software and hardware) predates this event

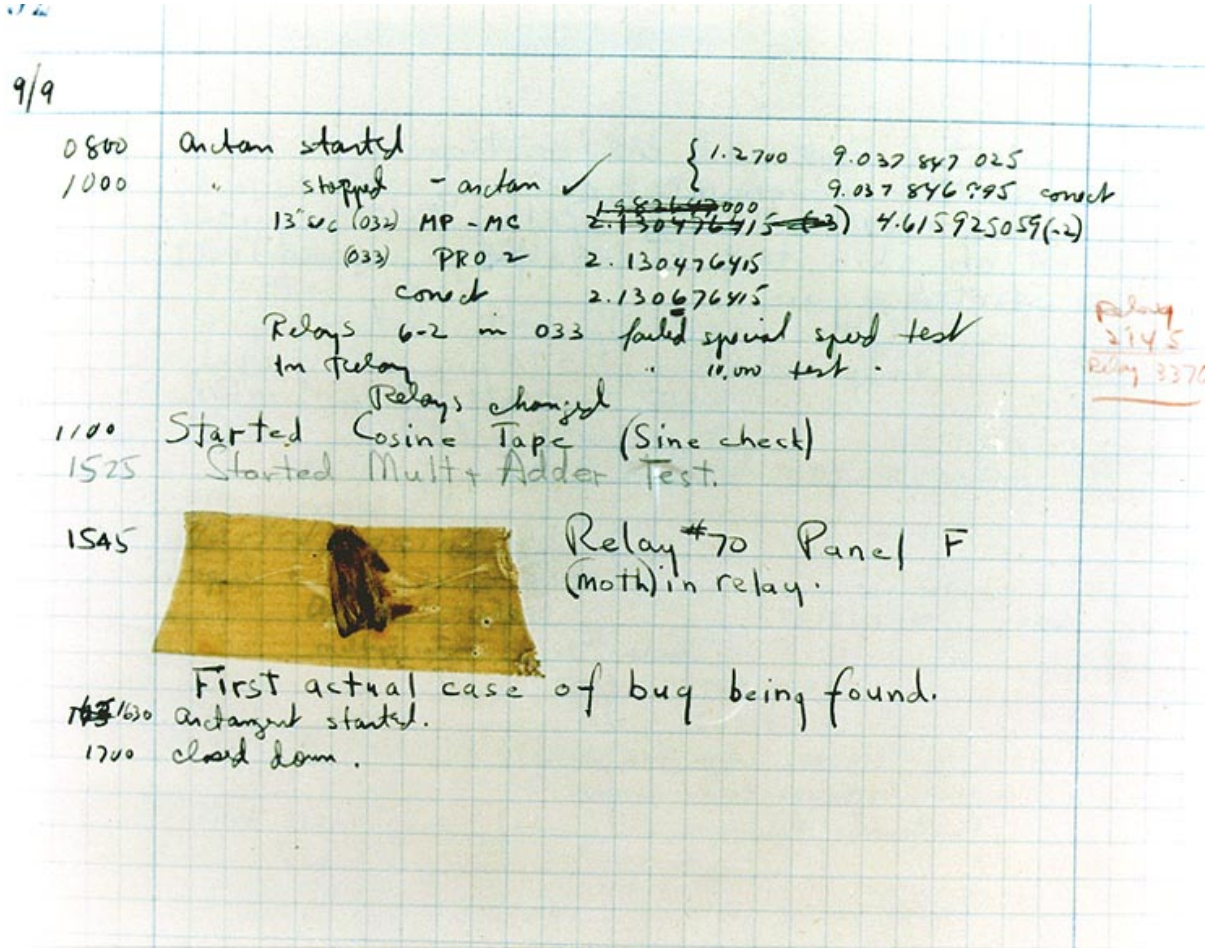


Image retrieved from https://en.wikipedia.org/wiki/File:H96566k.jpg on February 17th 2017
The image is in the public domain in the United States

Find the bug in this code

```
>> % Create a vector v of the form [10, 10^2, 10^3, ..., 10^n]
>> n = 10;
>> v = zeros(1, n);
>> for i = n
>>     v(i) = 10^i;
>> end
```

Find the bug in this code

```
>> % Create a vector v of the form [10, 10^2, 10^3, ..., 10^n]
>> n = 10;
>> v = zeros(1, n);
>> for i = n
>>     v(i) = 10^i;
>> end
```

Should be 1:n

Practice question

What will the value of the variable "var" be after executing the following code?

```
>> var = 0;
>> while var == var
>>     var = var + 1;
>>     if var >= 10 & var < 20
>>         continue
>>     end
>>     if var >= 10
>>         break
>>     end
>> end
```

(A) 0

(B) 2

(C) 20

(D) 21

(E) This `while` loop is an infinite loop

Practice question

What will the value of the variable "var" be after executing the following code?

```
>> var = 0;
>> while var == var
>>     var = var + 1;
>>     if var >= 10 & var < 20
>>         continue
>>     end
>>     if var >= 10
>>         break
>>     end
>> end
```

(A) 0

(B) 2

(C) 20

(D) 21

(E) This `while` loop is an infinite loop

Practice question

Consider the following functions:

```
function y = my_func1(x)
y = x + my_func2(x) + my_func3(x);
end
```

```
function y = my_func3(x)
y = x+2;
end
```

```
function y = my_func2(x)
y = my_func3(x) + x;
end
```

What will the value of the variable y be after executing the command `y = my_func1(3)`?

(A) 3

(B) 5

(C) 8

(D) 14

(E) 16

Practice question

Consider the following functions:

```
function y = my_func1(x)
y = x + my_func2(x) + my_func3(x);
end
```

```
function y = my_func3(x)
y = x+2;
end
```

```
function y = my_func2(x)
y = my_func3(x) + x;
end
```

What will the value of the variable y be after executing the command `y = my_func1(3)`?

(A) 3

(B) 5

(C) 8

(D) 14

(E) 16

Feedback on the class

On a piece of paper, write your feedback about E7 so far:

- Things that are going well
- Aspects of the class that could be improved

Suggestions of topics to discuss:

- **Changes since last feedback?**
- Lectures/discussions
- Lab assignments
- Lab sections
- Other resources (FAQs, bCourses Discussions and Pages, drop-in hours, etc.)