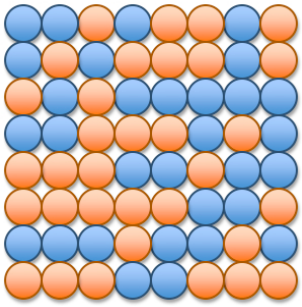


UNIT 3C
Using Loops and Conditionals

Finding the maximum

Required: a non-empty *list* of integers.

1. Set *max* equal to the first number in the *list*.
2. For each number *n* in the *list*:
   a. If *n* is greater than *max*, then set *max* equal to *n*.

Return: *max* as the maximum of the *list*.

Finding the max using Python

```
def findmax(numlist):
    max = numlist[0]
    for i in range(1,len(numlist)):
        n = numlist[i]
        if n > max:
            max = n
    return max
```

3

Alternate Version (Iterator)

```
def findmax2(numlist):
    max = numlist[0]
    for item in numlist:
      if item > max:
          max = item
    return max
```

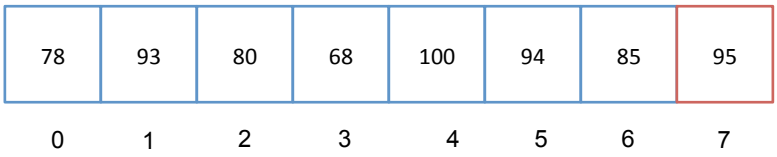
"For each item in numlist..."

Is there a redundant relational test done here?

4

Lists Are Mutable

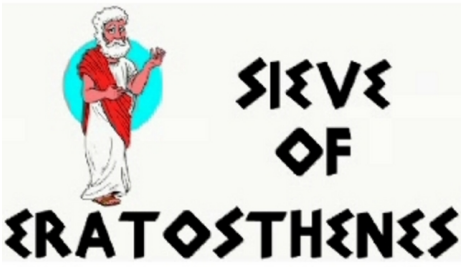
```
>>> scores
[78, 93, 80, 68, 100, 94, 85]
>>> scores.append(95)
>>> scores
[78, 93, 80, 68, 100, 94, 85, 95]
```




```
>>> scores.remove(68)
>>> scores
[78, 93, 80, 100, 94, 85, 95]
>>> scores.remove(42)
ValueError: list.remove(x): x not in list
```

what happens when there is more than one match?

5



A 2000 year old algorithm (procedure) for generating a table of prime numbers.

2, 3, 5, 7, 11, 13, 17, 23, 29, 31, …

A positive integer is "prime" if it is not divisible by any smaller positive integers except 1.

6

Sieve of Eratosthenes - Example

```
primelist = []
numlist = [2,3,4,5,6,7,8,9,10,11,12,13,
  14,15,16,17,18,19,20,21,22,23,24,25]

primelist = [2]
numlist = [3,5,7,9,11,13,15,17,19,21,23,25]

primelist = [2,3]
numlist = [5,7,11,13,17,19,23,25]

primelist = [2,3,5]
numlist = [7,11,13,17,19,23]       etc.
```


Sieve of Eratosthenes

To make a list of every prime number less than n:

1. Create a list *numlist* with every integer from 2 to n, in order. (Assume n > 1.)
2. Create an empty list *primelist*.
3. Copy the first number in *numlist* to the end of *primelist*. (It must be prime. Why?)
4. Iterate over *numlist* to remove every number that is a multiple of the most recently discovered prime number.
5. Halt if *numlist* is empty. Otherwise, go back to step 3.

Lists: Two Special Cases

```
values = []
```
This is the empty list (a list with length 0).

```
values = []
for i in range(1,10):
    values.append(i)
```
This is the list with the first 9 positive integers in order: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Starting the algorithm in Python

To make a list of every prime number less than n:
1. Create a list *numlist* with every integer from 2 to n, in order. (Assume n > 1.)
2. Create an empty list *primelist*.

```
def sieve(n):
    numlist = []
    for i in range(2,n+1):
        numlist.append(i)
    primelist = []
```

Continuing...

3. Copy the first number in *numlist* to the end of *primelist*. (It must be prime. Why?)

```
...
primelist.append(numlist[0])
...
```

Does this operation remove the first element from *numlist*?

Removing multiples of a prime

4. Iterate over *numlist* to remove every number that is a multiple of the most recently discovered prime number.

Where is the most recently discovered prime added to the **`primelist`** list?

`primelist[len(primelist)-1]`
(i.e. last element)

Removing multiples of a prime

4. Iterate over *numlist* to remove every number that is a multiple of the most recently discovered prime number.

How do we determine whether a number **x** is a multiple of **y**?

Use the modulo operator!

```
if x % y == 0:
    print("It's a multiple!")
```


Sifting:
Removing Multiples of a Number

```
def sift(numlist,k):
# remove all multiples of k from numlist
    index = 0
    while index < len(numlist):
        if numlist[index] % k == 0:
            numlist.remove(numlist[index])
        else:
            index = index + 1
    return numlist
```

Sifting Example

```
sift([1,2,3,4,6,7], 2)
```

list	index
[1,2,3,4,6,7]	0
[1,2,3,4,6,7]	1
[1,3,4,6,7]	1
[1,3,4,6,7]	2
[1,3,6,7]	2
[1,3,7]	2
[1,3,7]	3 (stop)

15

Removing multiples of a prime

Steps 3 & 4 together:

3. Copy the first number in *numlist* to the end of *primelist*.
4. Iterate over *numlist* to remove every number that is a multiple of the most recently discovered prime number.

```
primelist.append(numlist[0])
lastprime = primelist[len(primelist)-1]
numlist = sift(numlist, lastprime)
```

Removing multiples of a prime

5. Halt if *numlist* is empty. Otherwise, go back to step 3.

We need to repeat steps 3 and 4:

```
primelist.append(numlist[0])
lastprime = primelist[len(primelist)-1]
numlist = sift(numlist, lastprime)
```

until *numlist* is empty. How do we do this?

Repeating a task

Since we want to repeat a task, use a loop!
Since we don't know how many iterations are necessary, we will use a while loop.

```
while len(numlist) > 0 :
  primelist.append(numlist[0])
  lastprime = primelist[len(primelist)-1]
  numlist = sift(numlist, lastprime)
```

Repeating a task

Since we want to repeat a task, use a loop!
Since we don't know how many iterations are necessary, we will use a while loop.

```
while len(numlist) >= 1 :
    primelist.append(numlist[0])
    lastprime = primelist[len(primelist)-1]
    numlist = sift(numlist, lastprime)
```


Repeating a task

Since we want to repeat a task, use a loop!
Since we don't know how many iterations are necessary, we will use a while loop.

```
while len(numlist) != 0 :
    primelist.append(numlist[0])
    lastprime = primelist[len(primelist)-1]
    numlist = sift(numlist, lastprime)
```

Repeating a task

Since we want to repeat a task, use a loop!
Since we don't know how many iterations are necessary, we will use a while loop.

```
while numlist != []:
    primelist.append(numlist[0])
    lastprime = primelist[len(primelist)-1]
    numlist = sift(numlist, lastprime)
```


Final Algorithm in Python

```
def sift(numlist,k):
# remove all multiples of k from numlist
    index = 0
    while index < len(numlist):
        if numlist[index] % k == 0:
            numlist.remove(numlist[index])
        else:
            index = index + 1
    return numlist
```

Final Algorithm in Python (cont'd)

```
def sieve(n):
    numlist = []
    for i in range(2,n+1):
        numlist.append(i)
    primelist = []
    while len(numlist) > 0:
        primelist.append(numlist[0])
        lastprime = primelist[len(primelist)-1]
        # remove all multiples of lastprime
        # from numlist using sift function:
        numlist = sift(numlist, lastprime)
    return primelist
```


Some More List Operations

```
>>>scores = [78, 93, 80, 68, 100, 94, 85]
>>>80 in scores
True
>>>scores + scores
[78, 93, 80, 68, 100, 94, 85, 78, 93, 80, 68, 100, 94, 85]
>>>scores
[78, 93, 80, 68, 100, 94, 85]
>>>scores[1:3]
[93, 80]
>>>scores[1:7:2]
[93, 68, 94]
>>>scores.index(100)
4
>>>scores[-1]
85
```