

Debugging:

→ Removing the problems in a program

bug = problem in a program

→ Most basic technique is to add print statements in your program to print value of variables.

~~# Add numbers 1 to 5. It should be~~

~~sum = 0~~ for ~~i~~ in range(5): ~~1 + 2 + 3 + 4 + 5 = 15~~

sum = sum + i ~~1~~ Add a debug statement
print(i, "i", sum, sum)

→ Problem: program prints 10 we expect 15

→ Let the program tell you the problem.

→ Add a print statement to print the value of i and sum

print("i = ", i, "sum = ", sum)

→ You notice that last i is 4 so range(5) should be range(6)

Output:
i=0 sum=0
i=1 sum=1
i=2 sum=3
i=3 sum=6
i=4 sum=10

Dictionaries & Records

(19)

They are used to store in a single variable multiple values.

For example, a "student" value can be represented with

fname	→	First name
lname	→	Last name
address	→	address
major	→	major
ssid	→	student id

dictionary

```
student = { }
```

Add fields to student.

```
student['fname'] = 'George'
```

```
student['lname'] = 'Smith'
```

```
student['ssid'] = '999999'
```

```
student['major'] = 'Math'
```

```
print(student)
```

Output: { 'fname': 'George', 'lname': 'Smith' ... }

Also you could initialize student in the following way: (20)

```
student = { 'name': 'George', 'name': 'Smith',  
            'ssid': '999999', 'major': 'Math' }
```

(21) Example of using dictionaries

student.py
#input a list of students one by one,
#adds the students into a list
print the student names and grades in a table.

```
def Main():  
    print(" Read students in a class")  
    classlist = [] #initialize class to an empty list.  
    #read student info.  
    while True:  
        name name = input(" Enter Name: ")
```

```
classlist = [
```

George	95
Mary	92
John	94

...

```
}
```

```
20 | 4 |
```

grade just

name
left just

#print

for student in classlist:

George
Mary
John

```
95 name = student['name']
```

```
96 grade = student['grade']
```

```
97 print(name.ljust(20),
```

```
and str(grade).rjust(4))
```

convert to a
grade

```
grade = eval(input("grade:"))
```

```
student = { }
```

```
student['name'] = name
```

```
student['grade'] = grade
```

```
#add student to class list
```

```
classlist.append(student)
```

```
yesno = input("More students? yes/no")
```

```
if yesno == 'no':
```

```
break
```

print a table with the students in the class

#compute average in the class

sum = 0
for student in classList:

sum = sum + student['grade']

print("Average of class is: ", sum/len(classList))

Program Example

23

Guessing a number

C: Think of a number:

(think of 2)
just don't

say it

$$\text{mid} = \frac{1+5}{2} = 3$$

C: Is it larger than 5?

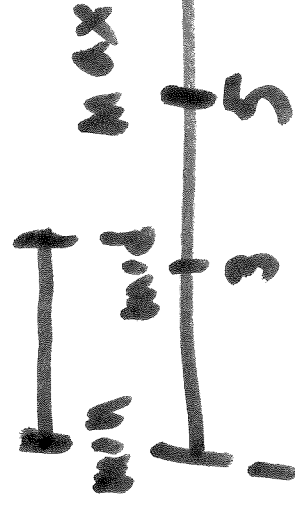
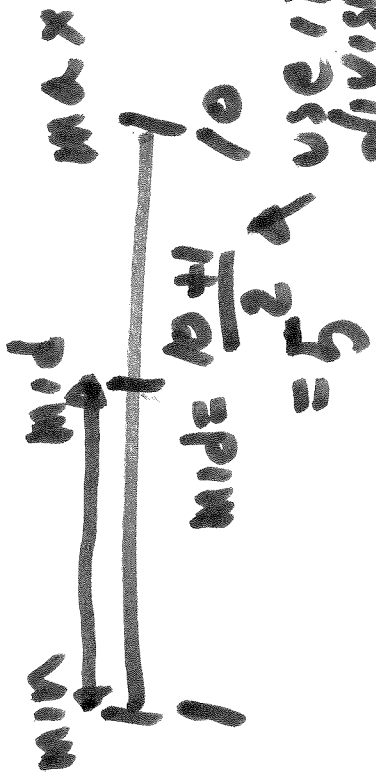
S: No

$$\text{mid} = \frac{1+5}{2} = 3$$

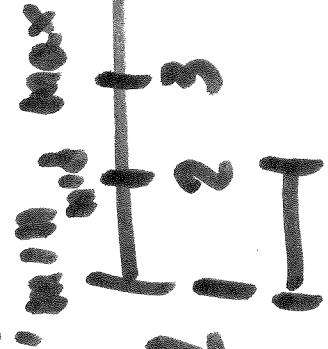
C: Is it larger than 3?

S: No

$$\text{mid} = \frac{1+3}{2} = 2$$



$$\frac{10}{2}$$



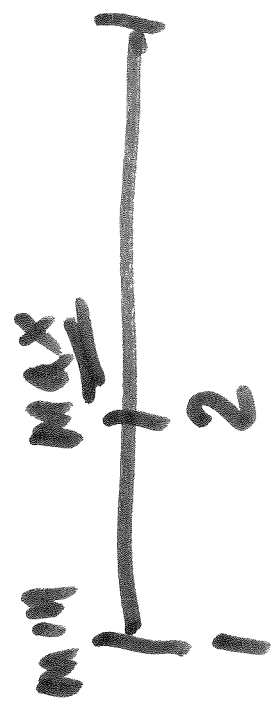
$$\frac{10}{2}$$

(24)

C: Is it larger than 2?

S: No.

$$mid = \frac{1+2}{2} = \frac{3}{2} = 1$$



$n/2$

C: Is it larger than 1? $min = 2$

S: Yes.



$I = \frac{n}{2}$

$min = max$

So the number 2 has to be 2

$$\frac{n}{2} = 1, \frac{n}{2} = 2, \frac{n}{2} = 1092$$

Implementing the guessing program

```
def guessingGame(start, end)
```

```
    min = start  
    max = end
```

Yes:

```
    print("Think of a number between  
          min, max, " and press enter")
```

```
    print input(" ")
```

int division

```
    while True:
```

```
        mid = (max + min) // 2
```

```
        ans = ("Is it larger than
```

```
               if ans == 'yes':
```

```
                   min = mid + 1
```

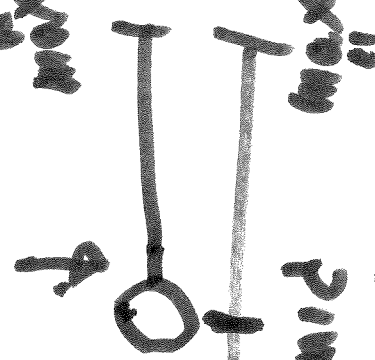
```
                   # does not include mid
```

```
               +  
               "mid")
```

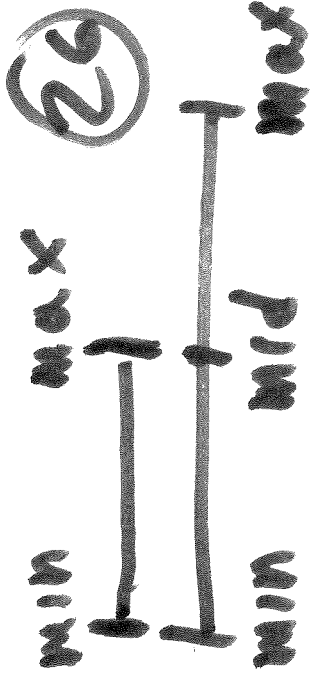
```
               "yes/no")
```

```
               # does not include mid
```

(25)




```
else:  
    max = mid
```



```
if min == max:  
    # we found the number  
    print("The number is", min)  
    break
```

```
def Main():  
    guessingGame(1, 100)  
  
Main()
```