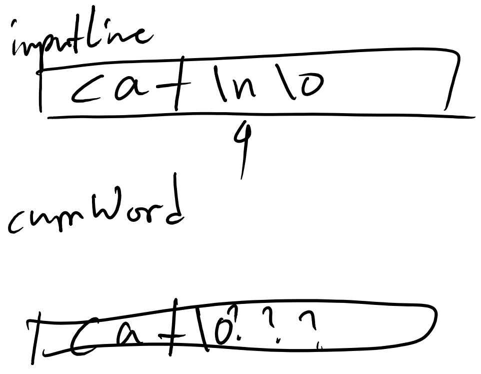
October 12, 2017

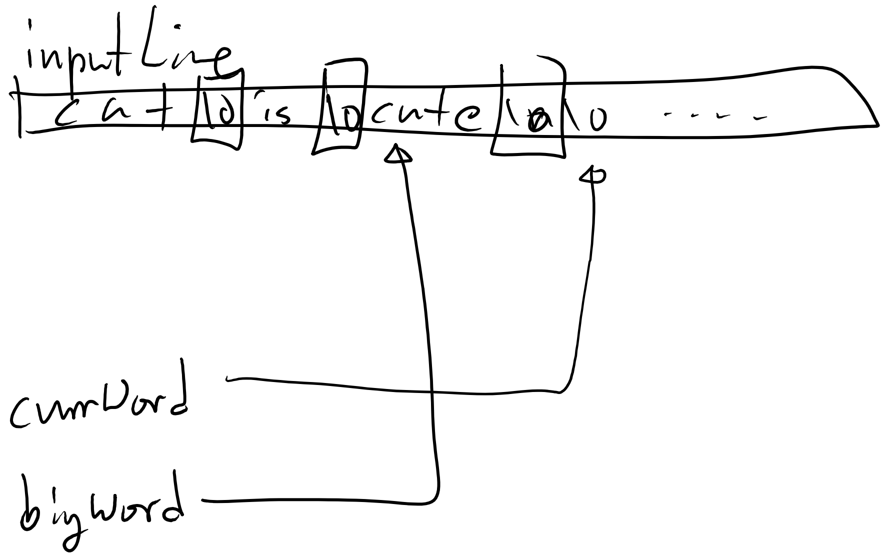
October 12, 2017 11:27 AM

```java
import ~~java~~.io.*;

public class Biggest
{
   public static void main(String[] args)
   {
      BufferedReader fileIn;
      PrintWriter fileOut;
      String inputLine;
      String currWord = "";
      String bigWord = null;
      int bigWeight = 0;
      int weight = 0;
      int index;
      int line = 0;

      try
      {
         fileIn = new BufferedReader(new FileReader("in.txt"));
         fileOut = new PrintWriter(new FileWriter("out.txt"));

         inputLine = fileIn.readLine();
         while ( inputLine != null )
         {
            index = 0;
            line++;
            bigWord = null;
            bigWeight = 0;

            while (index < inputLine.length())
            {
               weight = 0;
               currWord = "";
               while (index < inputLine.length() &&
                     (inputLine.charAt(index) != ' ' &&
                      inputLine.charAt(index) != '\t'))
               {
                  currWord += inputLine.charAt(index);
                  weight += inputLine.charAt(index);
                  index++;
               }

               if (weight > bigWeight)
               {
                  bigWord = currWord;
                  bigWeight = weight;
               }

               index++;
            }

            if (bigWord != null)
            {
               fileOut.print( "line " + line + "'s word: " + bigWord );
               fileOut.println();
            }

            inputLine = fileIn.readLine();
         }
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define LINE_MAX 2000

Int main ()
{
  char inputLine[LINE_MAX];
  char currWord[LINE_MAX];
  char bigWord[LINE_MAX];
  int bigWeight = 0;
  int weight = 0;
  int index;
  int line = 0;
  int indexCurrWord = 0;

  while( fgets( inputLine, LINE_MAX, stdin ) ) {
   index = 0;
   line++;
   bigWord[0] = '\0';
   bigWeight = 0;
   while( index < strlen( inputLine ) )
   {
                weight = 0;
                currWord[0] = '\0';
                indexCurrWord = 0;
                while (index < strlen(inputLine) &&
                      (inputLine[index] != ' ' &&
                       inputLine{index] != '\t') &&
                       inputLine[index] != '\n'
                {
                   currWord[indexCurrWord] +=
inputLine[index];
                   weight += inputLine[index];
                   index++;
                   indexCurrWord++;
                }
                currWord[indexCurrWord] = '\0';
                if (weight > bigWeight)
                {
                   strcpy( bigWord,  currWord );
                   bigWeight = weight;
                }

                index++;
   }

             if (bigWord[0] != '\0')
             {
             }
  }

  return 0;
}
```

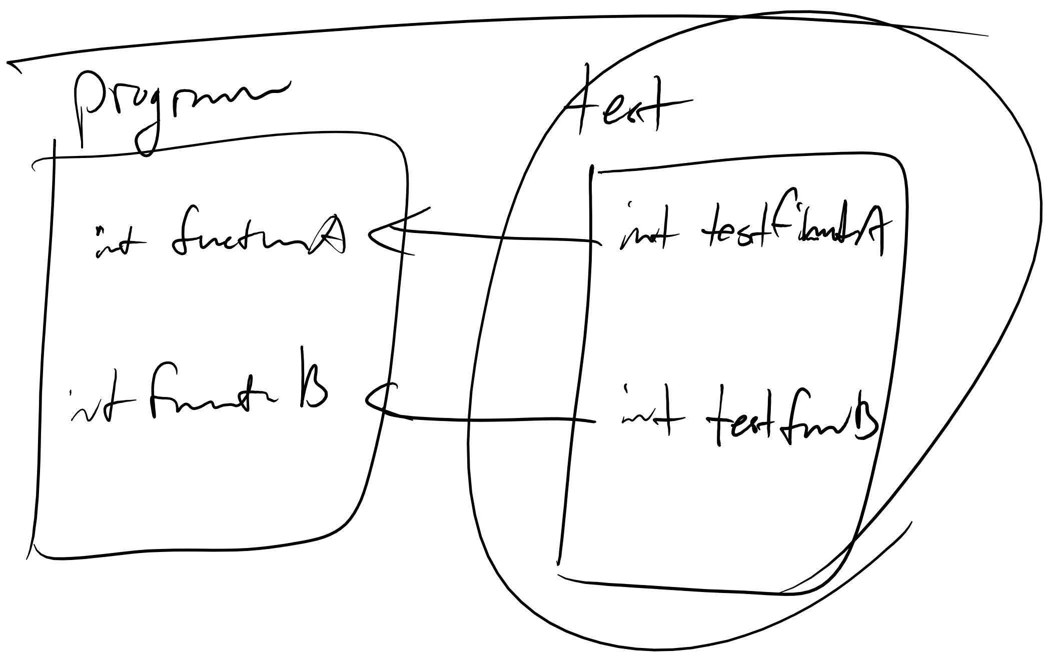
```
            fileIn.close();
            fileOut.close();
        }

        catch (IOException ioe)
        {
            System.out.println(ioe.getMessage());
            ioe.printStackTrace();
        }
    }
}
```





5/

```
5//
 \
  //
double power( double value , double factor);

                                   0.~~0~~

//
   board should not be NULL
//
int fcn( Board * board )
       board cannot be NULL


char * strdup ( . . . )


int fcn( Board * ptr ) {
    assert( ptr != NULL );
    if( ptr != NULL ) {



    }
```



int functionA

if(   )

assert (- -)
return

- function performs correctly at any given time

- to make sure it's working fine, make a test using scaffolding make a test using scaffolding,

- assert / if
  check your pre-/post- conditions
  are right,

↳ Design by Contract

- if you write code,
  and then write test,
  you may miss something

```
int checkSorted (int* array, int size)
{ ~~if(size ≥ 1) return 1;~~
  return 0;
}

void testCheckSorted() {
```

```
int data[]= { 1 , 0 } ;
assert (!checkSorted( NULL , 0)) ;
assert (!checkSorted(data , 0)) ;
assert (checkSorted(data , 1)) ;
assert ( checkSorted( data , 2)) ;
```

Test Driven Development
- Write test first
- make your function pass <u>all</u> tests.

```
//____________
(
//____________
```

```
printf("   . . -   "   ) ;
```

fflush(stdout); ✓