Written Final Exam

This exam is closed book - you are allowed only one page of notes (double- sided). If a question seems unclear - please write down any assumptions you feel are needed. If you think that there is a just-plain mistake/typo - check with an instructor.

Anywhere we ask you what will be printed out, if you think an error will be generated, you may write “error”. You do not need to write out what the whole error message would be.

1. (4  points)  You  are  working  collaboratively  with  a  friend  on  writing  some  code  for  a  project.  Your  friend  stayed  up  later  than  you  did  and  left  you  a  note,  saying  “Made  great  progress!  Put  all  my  changes  in  our  bitbucket  repo.”  Which  git  command  do  you  need  to  do  to  get  your  friend’s  latest  updates?

a. git  add

b. git  commit

c. git  pull

d. git  push

2. (4  points)  You’ve  made  some  changes  to  the  code  but  haven’t  done  any  git  operations  this  morning.  Your  friend  is  awake  now  and  would  like  you  to  share  your  changes.  In  which  order  would  you  do  these  git  operations?

a. add,  commit,  push

b. add,  push,  commit

c. add,  push,  pull

d. commit,  add,  push

e. commit,  add,  pull

f. pull,  push

3. (5  points)  You  have  been  part  of  an  open  source  project  from  its  founding.  A  new  leader  has  emerged  and,  well,  let’s  just  say  you  don’t  get  along.    The  leader  has  convinced  the  rest  of  the  core  group  to  kick  you  out.  The  project  is  still  remaining  an  open  source  project.  Which  of  the  following  things  will  you  not  be  able  to  do?  Circle  all  that  apply

a. Download  the  latest  version  of  the  code

b. Run  the  code

c. Modify  the  code

d. Get  your  modifications  incorporated  into  the  project’s  bitbucket

repository

e. Distribute  a  modified  version,  including  much  of  the  original  code  but  with  some  of  your  modifications,  through  a  different  bitbucket  repository

4. (5  points)  You  sense  a  commercial  opportunity  in  the  heartbleed  fiasco.  You  would  like  to  make  a  company  that  sells  a  better  version  of  the  OpenSSL  software.  You  plan  to  hire  people  to  do  extensive  code  reviews,  run  lots  of  automated  security  tests,  and  post  bounties  for  anyone  who  can  find  vulnerabilities  in  the  code  and  report  them  to  you.  This  will  cost  you  lots  of  money.  But  you  plan  to  earn  that  and  more  in  revenue  from  selling  your  software  rather  than  giving  it  away  for  free.  Your  customers  will,  for  a  fee,  get  access  to  your  better  version  of  the  software  and  will  have  the  right  to  run  it  but  not  to  modify  it  or  give  it  away  to  anyone  else.

Before  starting  your  company,  you  decide  to  check  out  the  license  under  which  the  existing  OpenSSL  software  is  released.    Would  it    be  better  for  your  business  plan  if  you  find  that  it  is  a  GPL  license,  or  a  BSD-­‐style  license?  Explain  why,  in  2-­‐3  sentences.

Assume  that  we  have  already  executed  the  following  code  (recall  that  """  is  the

delimiter  for  strings  that  are  on  more  than  one  line):

s = """<entry>

<id>tag:search.twitter.com,2005,1142881099</id>

<published>2009-01-23T20:04:53Z</published>

</entry> """

5. (4  points)  What  will  the  following  code  print  out?

print s.find(">")

6.  (4  points)  What  will  the  following  code  print  out?

print len(s.split(':'))

7. (5  points)  What  will  the  following  code  print  out?

print len(s.split('>')[2].split(':'))

8. (5  points)  Define  a  function  f  that  takes  a  list  of  strings  and  returns  a  list  containing  the  first  letter  of  every  word  that  contains  the  letter  z.  Make  it  pass  the  test  below.  Your  function  must  use  a  list  comprehension  to  create  the  list  it  returns,  or  use  map  and/or  filter.

def f(L):

test.testEqual(f(['Amazing', 'corny', 'zany']), ['A',

'z'])

9. (5  points)  Define  a  function  g  that  takes  a  list  of  strings  and  returns  a  list  of  them,  sorted  in  alphabetic  order  by  their  last  character.  Make  it  pass  the  test  below.

def g(L):

test.testEqual(g(['Amazing', 'corny', 'zanier']), ['Amazing', 'zanier', 'corny']) test.testEqual(g(['good', 'good on ya', 'good on you'], ['good on ya', 'good', 'good on you'])

10.  (5  points)  Fill  in  the  second  line  using  a  string  interpolation  (with  the  %

operator),  in  order  to  make  the  tests  pass.

def interp(x, y):

test.testEqual(interp(5, "sir"), "That is 5 in a row, sir. Congratulations!")

test.testEqual(interp(6, "your highness"), "That is 6 in a row, your highness. Congratulations!")

11. (5  points)  Fill  in  the  parameter  list  for  the  function  h  below  so  that  the  tests

pass.

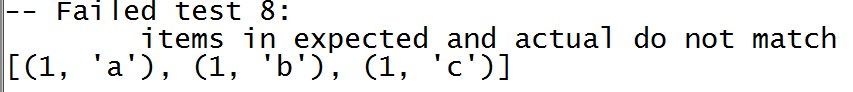
def h(x, y= 3, z = 4 ):

test.testEqual(h(1, 2), [1, 2, 4])

test.testEqual(h(1, z = 5), [1, 3, 5])

12. (5  points)  You  have  defined  a  function  enum.  It  takes  a  list  as  input  and  is  supposed  to  produce  a  list  of  tuples  that  number  the  original  items:  the  first  item  is  paired  with  1,  the  second  with  2,  etc.    Unfortunately,  your  code  isn’t  working  correctly  yet.  The  code  generates  the  output  shown  in  the  screenshot.  Rewrite  the  definition  of  enum  so  that  it  passes  the  test.

def enum(L):



def enum(L):

####write your new definition below

For  the  next  three  questions,  the  function  count\_guesses  has  been  defined  as  below,

slightly  modified  from  the  ps10  solutions.

def count\_guesses(next\_letter, guesses):

The  comment  string  defines  what  the  correct  output  should  be.  The  function  is

implemented  correctly,  so  any  test  you  write  should  pass.

13. (4  points)  In  order  to  write  test  cases  for  count\_guesses,  you  would  make:

a. Return  value  tests

b. Side  effect  tests

Briefly  justify  your  answer,  in  1-­‐2  sentences.

14.  (5  points)  Write  a  test  that  checks  that  the  right  thing  happens  when

next\_letter  is  among  the  guesses

test.testEqual(count\_guesses("a",  ["b",  "c",  "e",  "a",  "f"]),  4)

15. (5  points)  Write  a  test  that  checks  that  the  right  thing  happens  when

next\_letter  is  not  among  the  guesses.

test.testEqual(count\_guesses("a",  []),  None)

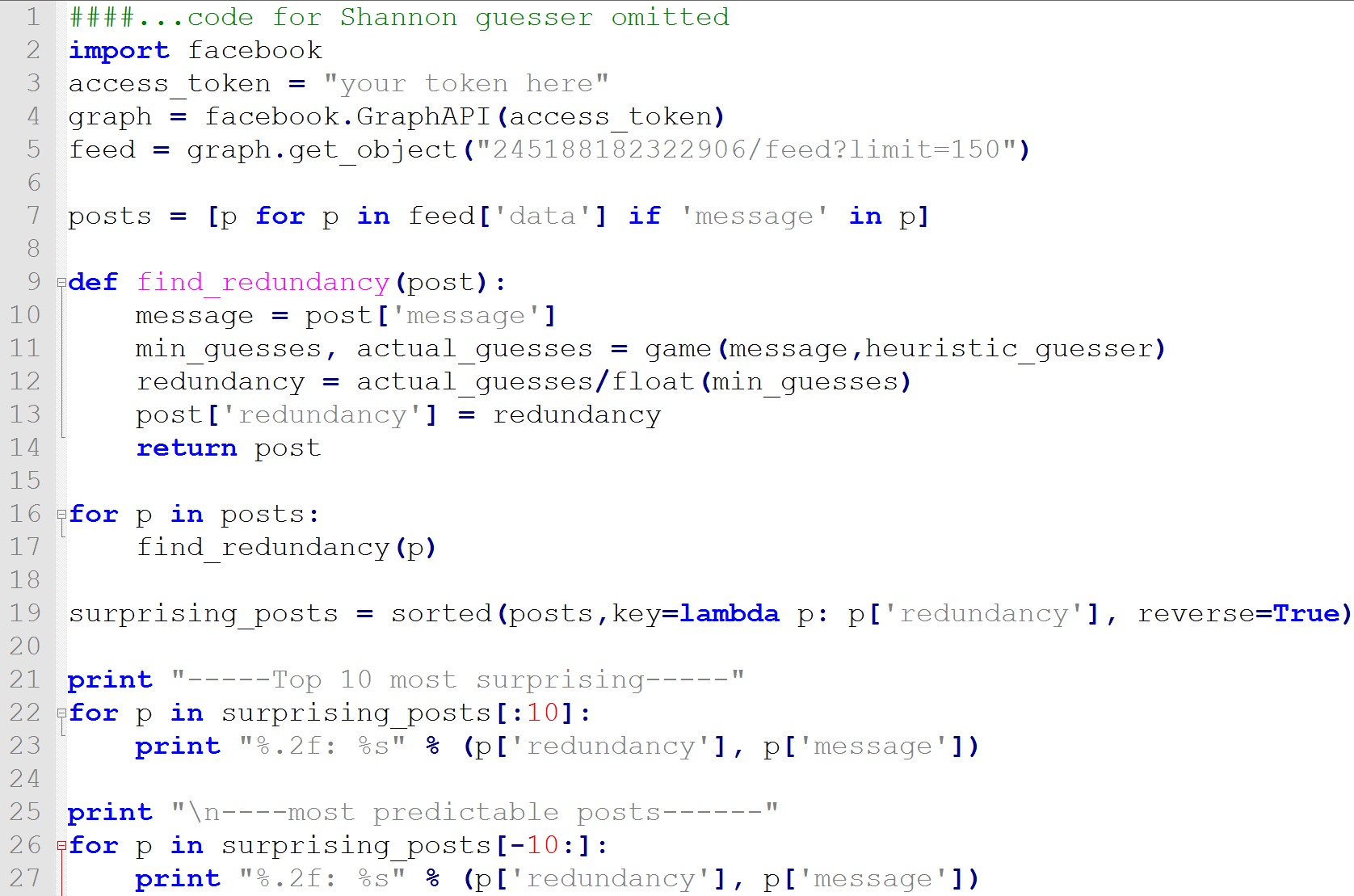
16.    (10  points)  Recall  that  we  discussed  in  class  that  after  a  sequence  of  capital  letters  in  a  text,  intuitively  it  would  have  made  more  sense  for  the  Shannon  guesser  to  guess  another  capital  letter.  As  one  way  to  do  that,  you  could  just  take  the  list  of  guesses  that  would  otherwise  be  made,  and  rearrange  them  to  put  all  the  capitals  first,  keeping  the  same  order  among  the  capitals  that  they  had  in  the  original  guess  list.  Define  a  function,  caps\_first,  which  does  that.  It  should  pass  the  test  below.

def  caps\_first(L):

test.testEqual(caps\_first(["A", "i", "K", "e"]), ["A", "K", "i", "e"])

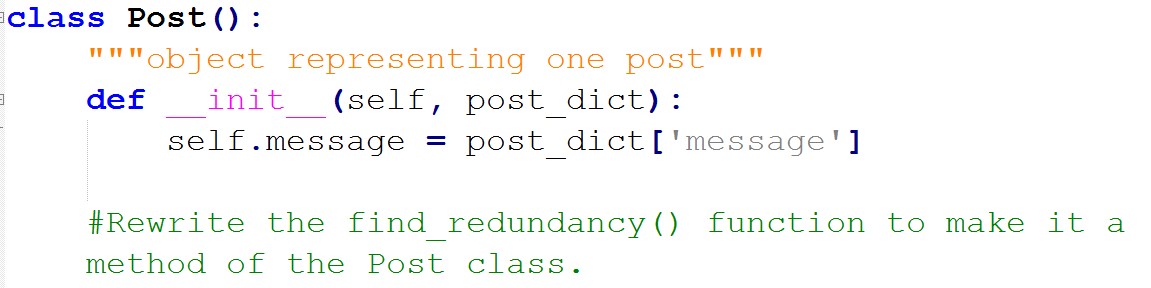
In  the  ps10  solutions,  we  provided  code  that  represented  each  Facebook  post  as  a  dictionary,  with  whatever  keys  were  included  in  the  json  dictionary  that  Facebook  returned,  plus  a  new  key  for  'redundancy'  that  our  code  generated.  The  relevant  code  from  ps10  is  below,  slightly  modified  and  simplified.  You  may  assume  that  game()  and  heuristic\_guesser()  are  defined  as  in  the  ps10  solutions,  but  not  shown  here.

In  ps9,  we  used  instances  of  a  class  Post  to  represent  Facebook  posts,  instead  of  dictionaries.  The  next  few  questions  walk  you  through  rewriting  the  ps10  solution  set  to  use  the  Post  class  instead  of  using  a  dictionary.



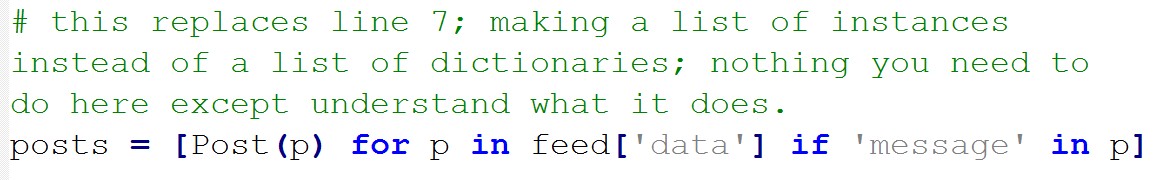
We  have  provided  a  skeleton  of  the  rewrite  of  the  code.  Follow  the  instructions  to

fill  in  various  lines.

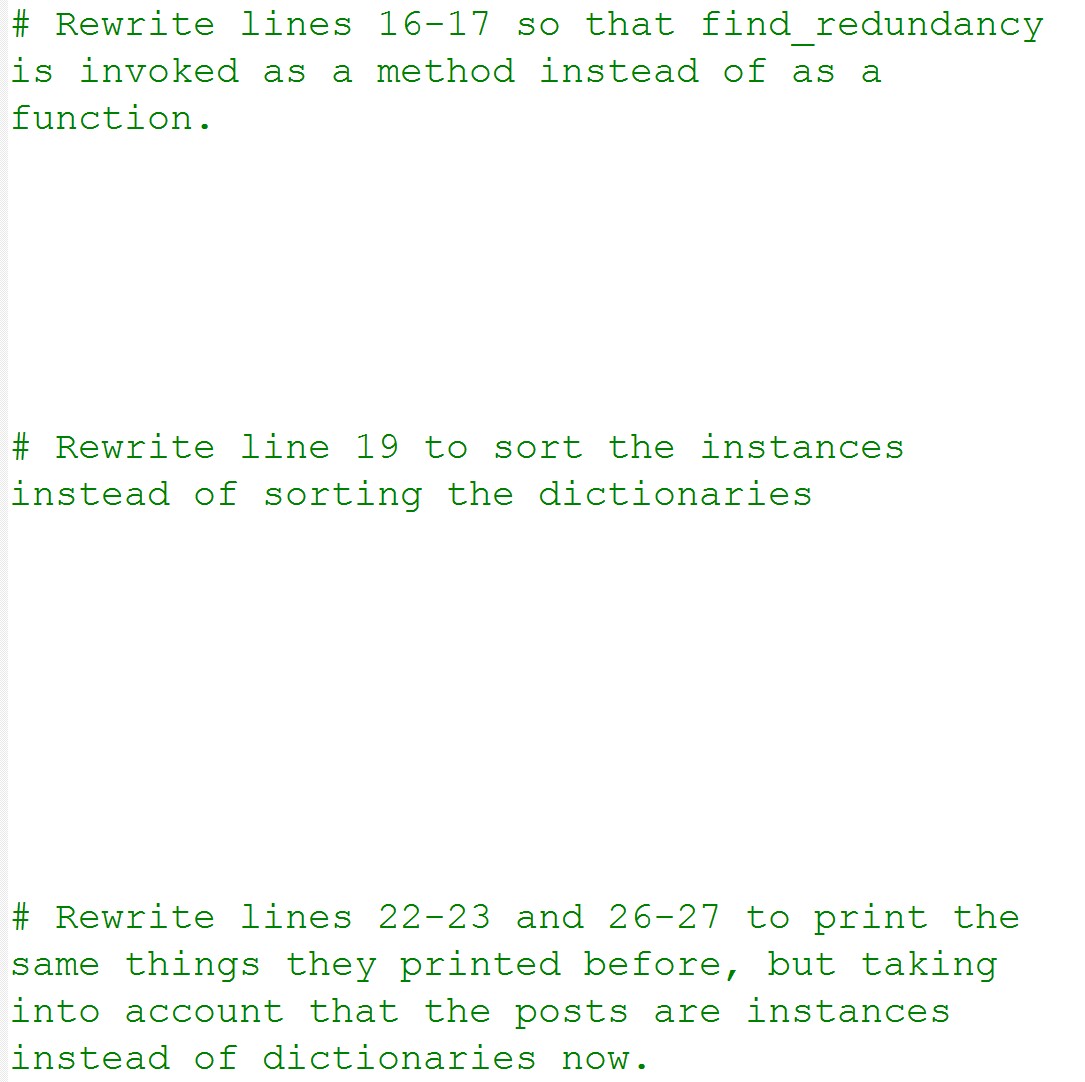


(5  points)

def find\_redundancy(self):



(5  points  each  for  the  next  three  questions)



surprising\_posts = sorted(posts,key=lambda p:

p.redundancy, reverse=True)