[OFFICIAL USE ONLY]

1 (15)	2 (18)	3 (20)	4 (23)	5 (7)	6 (13)	TOTAL (100)

CS101 Introduction to Programming 2011 Spring Final Examination

SECTION	STUDENT ID	NAME

※ Please check to see if you have all twelve pages in your test material (Total : 16 pages).
※ 시작하기 전에 반드시 페이지의 수를 확인 하십시오. (전체 : 16쪽)
※ Fill in your student identification number and name. Or you will lose 1 point for each missing piece of information.
※ 학번 및 이름을 정확히 기입하지 않을 경우, 각 실수 당 1점이 감점 됩니다.
※ **We will not answer questions about the problems during this exam**. If you think there is anything ambiguous, unclear or wrong about a problem, please write down your reasons and make necessary assumptions to proceed with the problem. We will take your explanation into consideration when grading.
※ **시험시간동안 질문을 받지 않습니다**. 만일 문제에 오류나 문제가 있을 경우, 왜 문제가 이상이 있다고 생각하는지에 대해서 기술하시면 되겠습니다. 또한 문제가 애매하다고 생각되는 경우 문제를 푸실 때 본인이 생각하는 가정을 함께 작성하셔서 문제를 푸시면 되겠습니다. 채점 시 가정 및 설명을 고려하도록 하겠습니다.

1. Answer each of the following questions according to the instruction. (15 points)

1-1. What is the output? (5 points)

```
print type("1" in "123") #(1 point)
```
<type ' '>

```
print type(1j) #(1 point)
```
<type ' '>

```
print type(False or 1) #(1 point)
```
<type ' '>

```
f = open("C:/planets.txt", "w")
print type(f)  #(1 point)
```
<type ' '>

```
print len('1\n2') #(1 point)
```


1-2. Fill out the box with "T" if the answer is true and with "F" for false. You also can choose "IDK" if you don't know the answer. If your answer is correct, then you'll get 1 point. Otherwise, you'll get -2 points. "IDK" or a blank will not be counted as any point.

#(5 points)

Statement	Answer
(Example) Variables defined outside of a function are called global variables.	T
Interpreter converts the input program to a numeric format that contains machine instructions.	
Merge Sort is an optimal sorting algorithm.	
A program is a set of operations that the computer can already perform.	
Every object has a ***type.***	
CS101 is about programming and computational thinking.	

1-3. What is the result of the following program?

#(2 points)

```
def f(data, end = 5, start = 0):
    return sum(data[start:end]) / (end-start)

d = [ 1, 2, 3, 4, 5 ]

print f(d, 2)
```

1-4. What is the result of the following program?

#(1 point)

```
from cs1graphics import *

book = Square(50)
book.setFillColor("black")
desk = book
desk.setFillColor("yellow")
photo = book
photo.setFillColor("green")

print desk.getFillColor()
```

1-5. What is the result of the following program?

#(2 points)

```
for a in [1]:
    for b in [1]:
        for c in [1]:
            if c==1:
                continue
            print 1,
        print 2,
    print 3,
print 4
```


2. Answer each question according to the instruction.

(18 points)

2-1. Write the output of following code.

#(6 points)

[Program Code]

```
l = []
s = 'mississipi'
l.append( len(s.split('s')) )
l.append( s.replace('s', 'x', 3) )
l.append( s.replace(s[4:], 'A') )
l.append( '.'.join(list(s)) )

m = ['%s'] * len(l)
print '\n'.join(m) % tuple(l)
```

[Output]

2-2. A function make_student_id takes two integer parameters, 'year' and 'number', and returns string of 8-digit student id number, which consists of 'year' and 'number'. Each parameter contributes 4 digits, and we assume that input parameter 'year' is integer from 2000 to 2011 and 'number' is integer from 0 to 9999. Fill in the blank (2-2) to get following output from the program code.

#(6 points)

[Program Code]

```
def make_student_id (year, number):

    return '          (2-2)          ' % (year, number)

print make_student_id( 2011, 1032 )
print make_student_id( 2007, 1 )
print make_student_id( 2010, 328 )
```

[Output]

```
20111032
20070001
20100328
```

Answer for blank (2-2):

2-3. A function get_domain takes one string parameter 'url', and returns string of its domain, which is the substring between 'http://' and the first slash '/' after 'http://'. Fill in the blank (2-3) to get following result from the program code not using split method.

#(6 points)

[Program Code]

```
def get_domain(url):
    (2-3)
    return domain

print get_domain('http://cs101.kaist.ac.kr/')
print get_domain('http://i.love.cs101.net/document_srl=840&listStyle=')
print get_domain('http://www.korea.com/A/B/C/cs101.cafe?ArticleNo=337')
```

[Output]

```
cs101.kaist.ac.kr
i.love.cs101.net
www.korea.com
```

Answer for blank (2-3):

```
def get_domain(url):
    # You should not use split method. Use find method instead.
    return domain
```


3. Answer each of the following questions according to the instruction.

(20 points)

3-1. Two words form a “anagram pair” if letters from one can be rearranged to spell the other; for example, “listen” and “silent”. The [Program Code] below is supposed to print all the sets of words that are anagrams in the order of the largest set of anagrams first, followed by the second largest set, and so on. Fill in the blanks so that when run, it will output [Result].

(8 points)

[Program Code]

```
# Description for ‘is_anagram’ function
# Return True if two words for a “anagram pair”; otherwise return False.
def is_anagram(str1, str2) :
    list1 = list(str1)
    list1.sort()
    list2 = list(str2)
    list2.sort()
    return (3-1-1)

def get_anagram_indices_of_a_word( a, b ) :
    index_list = []
    for i in range( len(b) ) :
        if is_anagram( a, b[i] ) :
            index_list.append(i)
    index_list.reverse()
    return index_list

def pop_words( a, b ) :
    sub_list = []
    for i in a :
        sub_list.append( b.pop(i) )
    return sub_list

def compare(list1, list2) :
    return cmp( (3-1-2) )

word_list = [ 'deltas', 'lasted', 'enlist', 'staled', 'generating',
'slated', 'silent', 'listen', 'greatening' ]
```

```
result_list = []
while len(word_list) > 0 :
    anagram_list = [ word_list.pop(0) ]
    index_list = get_anagram_indices_of_a_word( (3-1-3) )
    anagram_list += pop_words( (3-1-4) )

    result_list.append( anagram_list )

print word_list, result_list
result_list.sort(compare)
print result_list
```

[Result]

```
[]  [['deltas',  'slated',  'staled',  'lasted'],  ['enlist',  'listen',
'silent'], ['generating', 'greatening']]
[['generating',  'greatening'],  ['enlist',  'listen',  'silent'],  ['deltas',
'slated', 'staled', 'lasted']]
```

3-1-1. Fill in the blank.

(2 points)

3-1-2. Fill in the blank.

(2 points)

3-1-3. Fill in the blank.

(2 points)

3-1-4. Fill in the blank.

(2 points)

3-2. The [Program Code] below is supposed to take a list and return a new list with only the unique elements if there are duplicate elements in the list. Fill in the blanks so that when run, it will output [Result].

(7 points)

[Program Code]

```
# Description for 'has_duplicates' function
# Take a list and return True if there is any element that appears more than
# once; otherwise return False.
def has_duplicates( a_list ) :
    result_val = False
    for i in range( (3-2-1) ) :
        if a_list[i] in (3-2-2) :
            result_val = True
            break
    return result_val

# Description for 'remove_duplicates' function
# Take a list and return a new list with only the unique elements from the
# original.
def remove_duplicates( a_list ) :
    result_list = []
    (3-2-3)
    return result_list

test_list = [ 'deltas', 'lasted', 'salted', 'lasted', 'salted' ]

if has_duplicates( test_list ) :
    print remove_duplicates( test_list )
```

[Result]

```
['deltas', 'lasted', 'salted']
```

3-2-1. Fill in the blank.

(2 points)

3-2-2. Fill in the blank.

(2 points)

3-2-3. Fill in the blank.

(3 points)

3-3. Two words form a “metathesis pair” if you can transform one into the other by swapping two letters; for example, “converse” and “conserve.” The [Program Code] below is supposed to take two words and return True if two words form a “metathesis pair”; otherwise return False. Fill in the blanks so that when run, it will output [Result]. Fill in the blank.

(5 points)

[Program Code]

```
def is_metathesis( str1, str2 ) :
    result_val = False
    if len(str1) == len(str2) :
        (3-3)
    return result_val

print is_metathesis( 'converse', 'coversen' )
print is_metathesis( 'converse', 'conserve' )
print is_metathesis( 'converse', 'conserves' )
print is_metathesis( 'conserves', 'converse' )
print is_metathesis( ‘conberse’, ‘conserve’ )
```

[Result]

```
False
True
False
False
False
```

4. In this problem, we will implement a simple plagiarism detection program, which detects same or similar lines between two Python files. This code will print out the line if the line in two files are exactly same, otherwise it will print only the different parts in the "result.txt" file. All files in this problem are located in the same folder. The two copy candidate files and the main function file are given below.

(27 points)

[Copy candidate files]

	20100020.py	20101120.py
1	class HW():	
2	pass	class HW():
3		pass
4	h1 = HW()	HW1=HW()
5	h1.score = 100	HW1.score=100
6	print h1.score	
7		print HW1.score

[Main function file]

```
def catch_plagiarism(filename1, filename2):
    f1 = open(filename1,"r")
    f2 = open(filename2,"r")
    result = (4-1)
    while True:
        line1 = f1.readline()
        line2 = f2.readline()
        if (4-2) :
            break
        #if the line includes only "\n", fetch the next line
        while len(line1) == 1:
            line1 = f1.readline()
        while len(line2) == 1:
            line2 = f2.readline()

        #comparison between two lines
        if line1.strip() == line2.strip():
            print line1.strip()
        else:
            line1_list = line1.split("=")
            line2_list = line2.split("=")
```



```
            for i in range(len(line1_list)):
                if line1_list[i].strip() != line2_list[i].strip():
                    (4-3)

        if (4-2) :
            break
        #end of while-loop
    print "completed!"
    f1.close()
    f2.close()
    result.close()

catch_plagiarism("20100020.py","20101120.py")
```

4-1. Assume that you want to open "result.txt" file for writing. Fill in the blank.

(5 points)

4-2. The condition of termination in the while loop is when the "readline" function meets EOF(End of file) of any file. With the help reference below, fill out the termination condition part.

(7 points)

```
>>> help(file.readline)
Help on method_descriptor:

readline(...)
    readline([size]) -> next line from the file, as a string.

    Retain newline.  A non-negative size argument limits the maximum
    number of bytes to return (an incomplete line may be returned then).
    Return an empty string at EOF.
```

4-3. Fill in the blanks in order to write the output below to [result.txt].

(7 points)

[result.txt]

```
1.py:h1
2.py:HW1
1.py:h1.score
2.py:HW1.score
1.py:print h1.score
2.py:print HW1.score
```

4-4. What would be the result of the main program as shown in the Python shell?

(8 points)

5. The following program is a class which calculate the grade of student. It has some attributes and methods. Answer the question.

#(7 points)

```
grade_value = [80,65,50,30,20]

class Student(object):

    def __init__(self,student_id,mid,final):
        self.student_id = student_id
        self.mid = mid
        self.final = final
        self.total = mid + final
        self.avg = self.total/2

    def get_grade(self):

        if self.avg > grade_value[0]:
            return "A"
        elif self.avg > grade_value[1]:
            return "B"
        elif self.avg > grade_value[2]:
            return "C"
        elif self.avg > grade_value[3]:
            return "D"
        else:
            return "F"

    def __str__(self):
        return str(self.student_id)+" | "+self.grade

# main routine
st1 = Student(20110000,80,90)
st1.grade = st1.get_grade()
```

5-1. What is the output?

#(3 points)

```
print st1
```

5-2. How many attributes does the st1 object have?

#(4 points)

6. Define a class due to the following rules.

#(13 points)

- The class have to describe an animal among following list
  (Monkey, Lion, Zebra, Dog, Cat, Panda) <-- Class name
- The class has ***one or more attributes*** that describe the animal features such as the sound of animal or the skin color
- The class has ***one or more method*** that describe the animal actions such as running
  (You can use a 'pass' keyword and comments in order to write method.)
- The class have to define a ***constructor and __str__ method*** that return a string including the name and sound of animal
- You ***do not need to draw a form of animal*** using cs1graphics