

1 (25)	2 (25)	3 (25)	4 (25)	TOTAL (100)

[CS101] Introduction to Programming

2014 Fall - Final Examination

SECTION	STUDENT ID	NAME

- ※ Please check if you received all 16 pages of the test material.
- ※ 시작하기 전에 반드시 페이지의 수를 확인 하십시오.(전체 : 16쪽)

- ※ Fill in your student identification number and name. Otherwise you will lose 1 point for each missing piece of information.
- ※ 위의 정보(학번,이름)를 정확히 기입하지 않을 경우, 각 실수 당 1점이 감점 됩니다.

- ※ **TAs will not answer your questions about the exam.** If you think that there is anything ambiguous, unclear or wrong about a problem, please write the reasons and make necessary assumptions to solve the problem. We will take your explanation into consideration while 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. (25 points) Answer each of the following questions according to the instruction.

1-1. (8 Points) What is the result of the following program? (*all the commands are executed sequentially.)

```
>>> s = "mcdonalds"
```

```
>>> print s[2:]          #(1 point)
```

```
>>> l = [3.1415, 9, 10, "Jack","King"]
```

```
>>> print type(l)        #(1 point)
```

<type '_____'>

```
>>> print "%.2f" %(l[0])    #(2 points)
```

```
>>> l.append([1, 2, 3])
```

```
>>> print len(l)          #(2 points)
```

```
>>> q = s
```

```
>>> print not s == q       #(2 points)
```

1-2. (7 Points) Fill out the answer field with "T" if the answer is **true** or "F" **otherwise**. You can also write "IDK" if you don't know the answer. You will get one point for each correct answer. Otherwise, you will lose two points. "IDK" or a blank will not be counted as any point.

Statement	Answer
<i>(Example) The first character of a name should not be a digit.</i>	T
Variables defined outside of a function are called global variables .	
A function's parameters are local variables .	
pop() removes and returns the first element of a list.	
Strings are mutable.	
A function can return only one object .	
Arguments are mapped to parameters one-by-one, right-to-left .	
Whenever an object is created, the constructor of the object class is called.	

1-3. (4 Points) What is the result of the following program?

```
def avg(data, start = 0, end = None):
    if not end:
        end = len(data)
    return sum(data[start:end]) / float(end-start)

d = [ 6, 7, 8, 9, 10 ]
print "%d" % avg(d)          #(2 points)
print "%d" % avg(d,2)        #(2 points)
```


1-4. (2 Points) Which of the following statements are incorrect? (Select 2 choices)

```
from math import sin, cos, pi

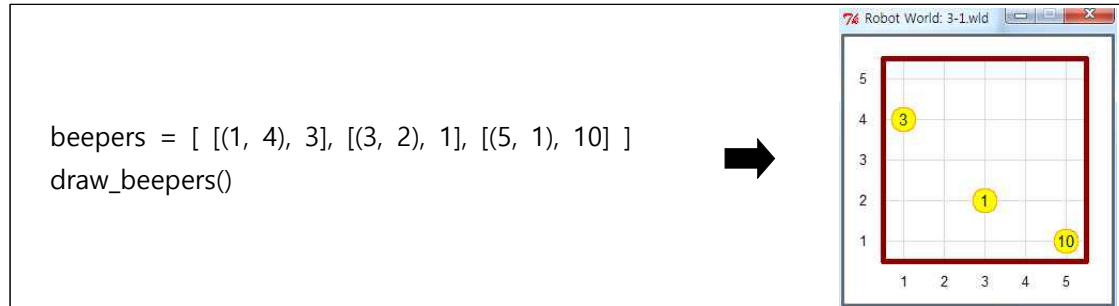
print pi                #----- (1)
print math.pi           #----- (2)
print cos(2*pi)         #----- (3)
print tan(pi / 4)       #----- (4)
```

1-5. (4 Points) What is the result of the following program?

```
for x in range(5):
    for y in range(x):
        if x%2 == 0:
            continue
        if y == 2:
            break
    print (y+1)*"#"
```

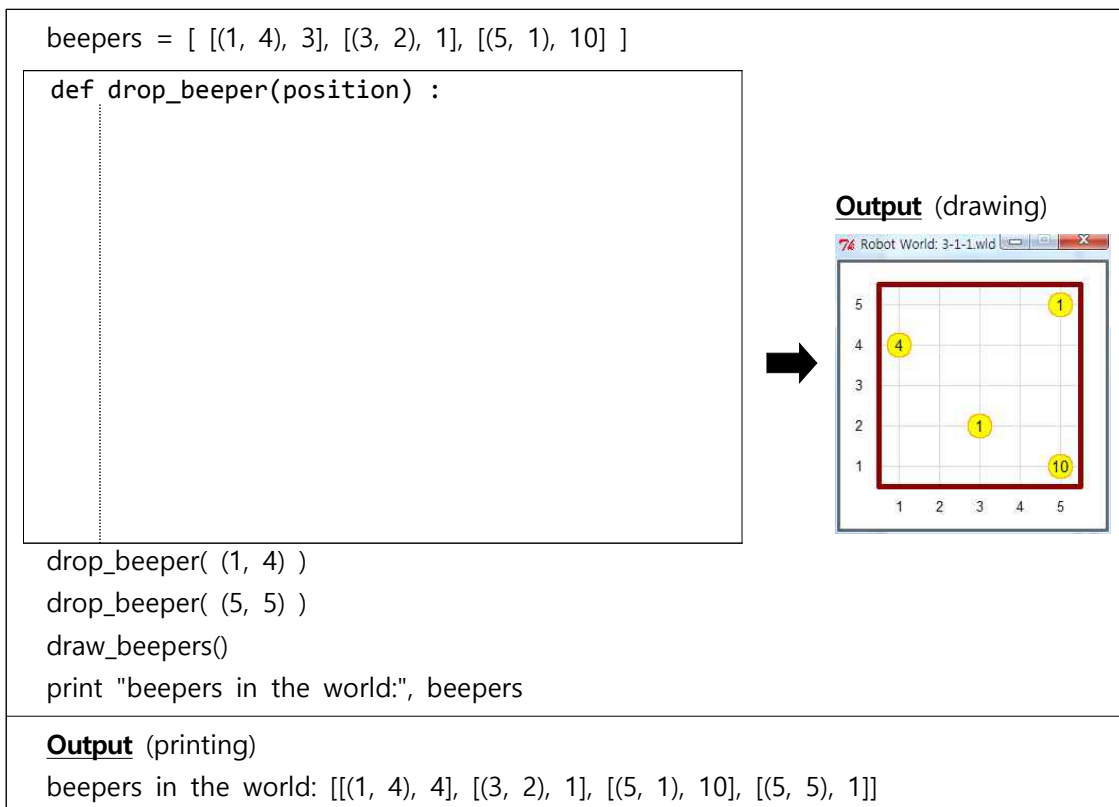

2. (25 points) Answer each question according to the instruction.

2-1. (8 points) A 5×5 world is drawn on a Canvas. A global variable, "beepers" is a list for all beepers shown in the world. Each beeper is represented as a list of a tuple that represents the x and y coordinates of the beeper in the world, and an integer value that represents the quantity of the beeper. A function, "draw_beepers" re-draws all beepers on the Canvas as shown below:



With the given variable and function, answer the following sub-questions.

2-1-1. (4 points) Implement a function, "drop_beeper", which adds a beeper to the world so that the following code results in the following outputs (i.e., drawing and printing). The function takes the position of the beeper being dropped as a tuple. It is assumed that only a valid input is given to the function. (※ The function must not be implemented only for the given output.)



2-1-2. (4 points) Implement a function, *"merge_beeper"* so that the following code results in the following outputs (i.e., drawing and printing). The function takes a list of new beepers (in the same structure as *"beepers"*), and merges it with *"beepers"* by using the *"drop_beeper"* function. (※ **The function must call "drop_beeper"**, instead of directly manipulating the *"beepers"* list. The function should be able to merge any list of beepers.)

```
beepers = [ [(1, 4), 3], [(3, 2), 1], [(5, 1), 10] ]
def drop_beeper(position) :
    # assume that this function has been correctly
    implemented
```

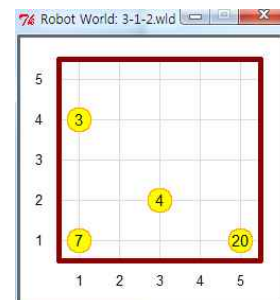
```
def merge_beeper(new_beepers) :
```

```
merge_beeper([ [(3, 2), 3], [(1, 1), 7], [(5, 1), 10] ])
draw_beepers()
print "beepers in the world:", beepers
```

Output (printing)

beepers in the world: [[(1, 4), 3], [(3, 2), 4], [(5, 1), 20], [(1, 1), 7]]

Output (drawing)



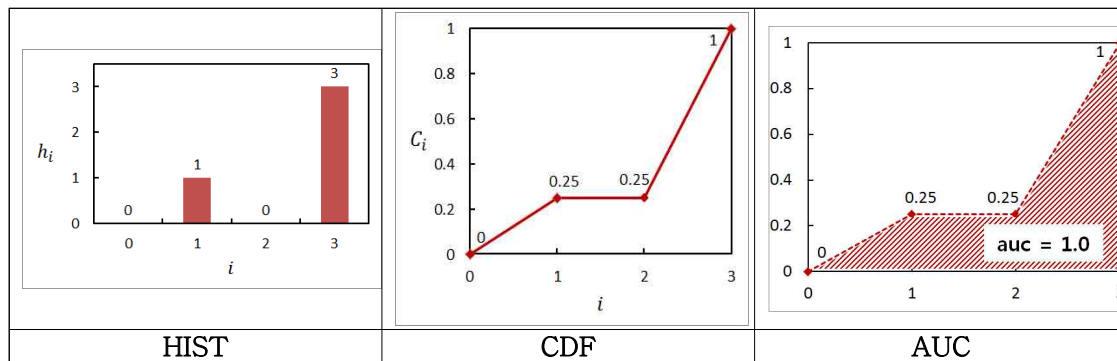
2-2. (12 points) The following program is to calculate the area under the curve of a cumulative distribution function from sample data. (Refer the graph below.)

```
# Do not answer here
def HIST(samples) :
    # answer in 3-2-1
def CDF(hist) :
    # answer in 3-2-2
def AUC(cdf):
    # answer in 3-2-3
samples = [5, 3, 7, 1, 7, 2, 1, 3]
print "histogram:", HIST(samples)
print "cdf:", CDF(HIST(samples))
print "auc:", AUC(CDF(HIST(samples)))
```

Output (printing)

```
histogram: [0, 2, 1, 2, 0, 1, 0, 2]
cdf: [0.0, 0.25, 0.375, 0.625, 0.625, 0.75, 0.75, 1.0]
auc: 3.875
```

For this, you are asked to complete three functions, "HIST", "CDF", and "AUC". Fill in the blanks in the following sub-questions, 3-2-1, 3-2-2, and 3-2-3.



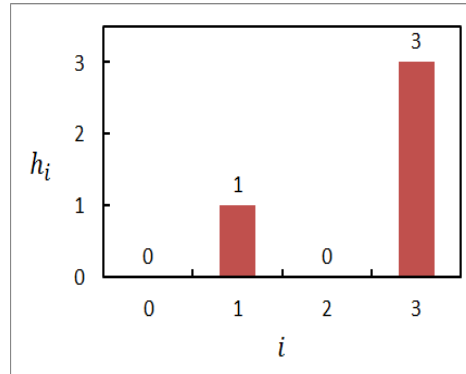
2-2-1. (4 points) The function, "HIST" computes a histogram of samples. Histogram is an estimate of the probability distribution of the samples (sample data). It is assumed that x-axis is from zero to the maximum value of the samples, and the length of the intervals on the x-axis is all one. By the assumption, a histogram h of the samples $S(x)$ is computed as follows:

$$h_i = \sum_x f\{S(x) = i\}, i = 0, \dots, m,$$

where m is the maximum value of the samples and

$$f\{X\} = \begin{cases} 1, & \text{if } X \text{ is true} \\ 0, & \text{otherwise.} \end{cases}$$

Both input and output are a list of integers. When [3, 1, 3, 3] is given, for example, the function "HIST" should return [0, 1, 0, 3] that can be used to draw a histogram like:

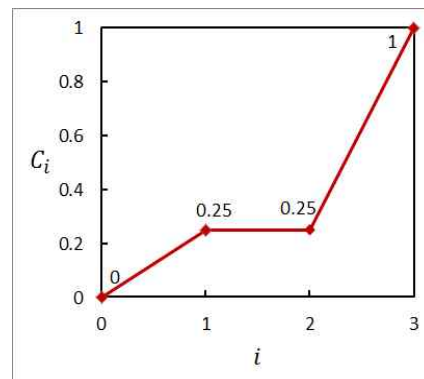
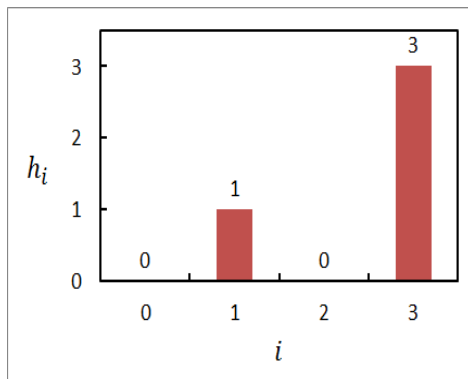


```
def HIST(samples) :
    hist = [0] * (max(samples) + 1)
    for s in samples :
        
    return hist
```

2-2-2. (4 points) The function, "CDF" computes an estimation of the cumulative distribution C from the histogram h as follows:

$$C_i = \frac{1}{n} \sum_{j=0}^i h_j, i = 0, \dots, m,$$

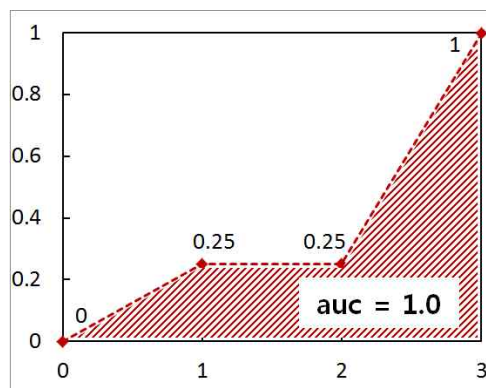
where m is the maximum value of the samples, and n is the number of the samples. The input of the function is a list of integers, and the output is a list of floats. When [0, 1, 0, 3] is given, for example, the function, "CDF" should return [0.0, 0.25, 0.25, 1.0] as follows (as shown on the graph at the right side):



```
def HIST(samples) :
    # assume that this function has been correctly implemented

    def CDF(hist) :
        cdf = []
        for i in range(len(hist)) :
            cdf.append(float(sum())/sum(hist))
        return cdf
```

2-2-3. (4 points) The function, "AUC" calculates the area under the curve of the cumulative distribution function (auc). The input of the function is a list of floats and the output is a float. When $[0.0, 0.25, 0.25, 1.0]$ is given, for example, the AUC function should return 1.0 as the auc value as shown on the following graph:




```

def HIST(samples) :
    # assume that this function has been correctly implemented
def CDF(hist) :
    # assume that this function has been correctly implemented

def AUC(cdf) :
    area = 0
    for i in range(1, len(cdf)) :
        area += 
    return area

```

2-3. (5 points) Assume that you have a global variable, *"team"*, which is a list of the WC2014Team objects that you worked on for the programming homework #4. You are asked to sort the list based on the total goal score in **descending order**. However, you are not permitted to directly access the attributes of the objects. Instead, you are given with a function, *"sort_team"* to sort the team list based on the total goal score **in ascending order**. The input of the function is a list of WC2014Team objects and the output is none. Using the function, sort the global variable, *"team"* based on the total goal score in descending order.

(※ Hint: You don't need to define new functions or new variables)

3. (25 points) You are asked to complete a puzzle game that shuffles the tiles of an original image. An example of shuffling an image is shown below. Please answer each question according to the instruction. The program to complete is shown on the next page.

* A 'tile' is a square shaped, and 'tile_size' is the length of a side of the tile. Therefore, if 'tile_size' is 40 then an original image with the size 640*480 is divided into 192 tiles (the size of each tile is 16*12).

* We assume that the width and height of an original image can be divisible by 'tile_size', i.e., if the size of the image is 640*480 then 'tile_size' is 10, 20 or 40 rather than 17 or 22.



3-1. (8 points) Please write code to set the pixels of the new tile with the pixels of a tile (its location is specified with tile_x and tile_y) in the input image received via the 'img' parameter of the 'get_tile' function.

3-2. (9 points) Please write code to fill 'new_img' with the tiles in the list, 'tiles'. You must call the 'set_tile' function.

3-3. (8 points) There are three errors in the program including the one shown below. Please find other errors and correct them properly.

	Error code	Correct code
Example	from cs1media import &	from cs1media import *
1		
2		

```

from cs1media import &
from random import *

def get_tile(img, tile_size, tile_x, tile_y):
    tile = create_picture(tile_size, tile_size)
    for x in range (tile_size):
        for y in range (tile_size):
            
        return tile

def set_tile(img, tile_size, w, h, tile):
    for x in range (tile_size):
        for y in range (tile_size):
            img.set(tile_size*w+x, tile_size*h+y, tile.get(x,y))

#main procedure
img = load_picture( "pororo.jpg" )
width, height = img.size()

new_img = create_picture(width, height)

tiles = ()
tile_size = 40
for w in range (width/tile_size):
    for h in range (height/tile_size):
        tile = get_tile(img, tile_size, w, h)
        tiles.append(tile)

random.shuffle(tiles)

new_img.show()

```

4. (25 points) Please complete a program that handles information about currency exchange rates. The currency information needs to be accessed from Federal Reserve System, which is the central banking system of the United States, <http://www.federalreserve.gov> (see the screen shot and HTML source of the Web page below). We assume that the corresponding Web page is downloaded into a local file (e.g., Weekly_exchange_rates.htm). Exchange rates are shown in US Dollar. For example, on Nov. 17, 1 US Dollar was equal to 0.8718 Australian Dollar.

'Weekly_exchange_rates.htm'						
Opened in Web Browser				Opened in Text Editor (key parts)		
Current Release				...		
Release Date: November 24, 2014				<thead>		
				<tr>		
				<th id="a1" colspan="1">COUNTRY</th>		
				<th id="a2">CURRENCY</th>		
				<th id="a3"> Nov. 17 </th>		
				<th id="a4"> Nov. 18 </th>		
				<th id="a5"> Nov. 19 </th>		
				<th id="a6"> Nov. 20 </th>		
				<th id="a7"> Nov. 21 </th>		
				</tr>		
				</thead>		
				<tr>		
				<th id="r1" headers="a1"> *AUSTRALIA </th>		
				<td headers="a2 a1 r1">DOLLAR </td>		
				<td headers="a3 a1 r1"> 0.8718</td>		
				<td headers="a4 a1 r1"> 0.8726</td>		
				<td headers="a5 a1 r1"> 0.8623</td>		
				<td headers="a6 a1 r1"> 0.8610</td>		
				<td headers="a7 a1 r1"> 0.8674</td>		
				</tr>		
				<tr>		
				<th id="r1" headers="a1"> BRAZIL </th>		
				<td headers="a2 a1 r1">REAL </td>		
				<td headers="a3 a1 r1"> 2.6016</td>		
				<td headers="a4 a1 r1"> 2.5954</td>		
				<td headers="a5 a1 r1"> 2.5744</td>		
				<td headers="a6 a1 r1"> 2.5718</td>		
				<td headers="a7 a1 r1"> 2.5247</td>		
				</tr>		
				...		

COUNTRY	CURRENCY	Nov. 17	Nov. 18	Nov. 19	Nov. 20	Nov. 21
*AUSTRALIA	DOLLAR	0.8718	0.8726	0.8623	0.8610	0.8674
BRAZIL	REAL	2.6016	2.5954	2.5744	2.5718	2.5247
CANADA	DOLLAR	1.1308	1.1305	1.1349	1.1311	1.1237
CHINA, P.R.	YUAN	6.1242	6.1208	6.1190	6.1230	6.1244
DENMARK	KRONE	5.9795	5.9393	5.9317	5.9410	6.0033
*EMU MEMBERS	EURO	1.2447	1.2534	1.2548	1.2526	1.2394
HONG KONG	DOLLAR	7.7548	7.7547	7.7550	7.7557	7.7572
INDIA	RUPEE	61.8200	61.7200	61.9800	61.9200	61.7600
JAPAN	YEN	116.4400	116.7600	117.7000	118.1400	117.7400
MALAYSIA	RINGGIT	3.3460	3.3515	3.3590	3.3650	3.3525
MEXICO	PESO	13.5870	13.5450	13.6070	13.6170	13.6255
*NEW ZEALAND	DOLLAR	0.7935	0.7941	0.7859	0.7852	0.7886
NORWAY	KRONE	6.7882	6.7228	6.7877	6.7641	6.7966
SINGAPORE	DOLLAR	1.2980	1.2962	1.3057	1.3008	1.2981
SOUTH AFRICA	RAND	11.1150	11.0245	11.0665	10.9625	10.9360
SOUTH KOREA	WON	1098.1200	1096.6400	1112.0400	1111.9600	1111.0500
SRI LANKA	RUPEE	130.9500	131.0000	131.0000	131.0000	131.0000
SWEDEN	KRONA	7.4302	7.3605	7.3855	7.4098	7.4754
SWITZERLAND	FRANC	0.9653	0.9583	0.9572	0.9596	0.9696
TAIWAN	DOLLAR	30.7000	30.7300	30.9000	30.8900	30.8900
THAILAND	BAHT	32.7800	32.7300	32.8000	32.8100	32.7800
*UNITED KINGDOM	POUND	1.5645	1.5648	1.5670	1.5683	1.5672
VENEZUELA	BOLIVAR	6.2842	6.2842	6.2842	6.2842	6.2842
Memo:						
UNITED STATES	DOLLAR					
1) BROAD	JAN97=100	107.8939	107.6877	107.9326	107.9499	107.9695
2) MAJOR CURRENCY	MAR73=100	82.8565	82.6269	82.8132	82.8474	83.0093
3) OITP	JAN97=100	134.3157	134.1459	134.4529	134.4491	134.2943
ND = No data for this date.						

User-defined object	Date
Description	Stores date information and provides string representation and equality test
Example	<pre>>>> dateInstance = Date('<th id="a3"> Nov. 1 </th>\n') >>> print dateInstance Nov. 1 >>> dateInstance == 1101 True >>> dateInstance == 111 False</pre>
<pre>def parse(oneLine): return oneLine.split('>')[1].split('<')[0].strip() months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'] def makeDateToInt(mon, date): for i in range(len(months)): if mon == months[i]: if int(date) < 10: return int("%d0%s" % (i + 1, date)) return int("%d%s" % (i + 1, date)) class Date(object): def __init__(self, dateline): temp = parse(dateline).split(' ') self.date = makeDateToInt(temp[0], temp[1]) # data type of int def <input type="text" value="# 4-1-1"/> (self): return <input type="text" value="# 4-1-2"/> def <input type="text" value="# 4-2-1"/> (self, rhs): return <input type="text" value="# 4-2-2"/></pre>	

4-1. (7 points) Complete the special method that converts the date information to a string.

```
def  (self):
    return 
```

4-2. (7 points) Complete the special method that tests the equality of the parameter, 'rhs' with the attribute, 'date'. The method returns 'True' if they are the same, 'False' otherwise.

```
def          (self, rhs):

    return
```

4-3. (6 points) Complete the method, *'change'* of the object, *'CurrencyRate'*. Given exchange date (*'dateInt'*), country name (*'fromCountryName'*), and amount of money (*'fromMoneyAmount'*) of the source country, the method returns the amount of money in the currency of the target country (*'toCountryName'*) on the given date.

User-defined object	CurrencyRate
Description	Stores weekly currency exchange rates and provides a method, <i>'change'</i> that calculates amount of money in the currency of the target country
Example	<pre>>>> currency = CurrencyRate() >>> currency.insertDate(Date("Nov", "17")) >>> currency.insertDate(Date("Nov", "18")) >>> currency.insertCountry(["AUSTRALIA", "DOLLAR"]) >>> currency.insertCountry(["BRAZIL", "REAL"]) >>> currency.insertExchangeRate(["AUSTRALIA", 0.8718, 0.8726]) >>> currency.insertExchangeRate(["BRAZIL", 2.6016, 2.5954]) >>> print currency.dates[0] Nov. 17 >>> currency.countries [['AUSTRALIA', 'DOLLAR'], ['BRAZIL', 'REAL']] >>> currency.exchangeRates [['AUSTRALIA', 0.8718, 0.8726], ['BRAZIL', 2.6016, 2.5954]] >>> currency.change(1118, "AUSTRALIA", 10000, "BRAZIL") 29743.29589731836 >>> currency.change(1118, "BRAZIL", 10000, "AUSTRALIA") 3362.1021807813822 >>> currency.change(1118, "AUSTRALIA", 10000, "AUSTRALIA") 10000.0</pre>

Method of 'CurrencyRate'	change
Input	<i>dateInt</i> (int): date when the currency exchange occurs <i>fromCountryName</i> (str): name of the source country <i>fromAmountOfMoney</i> (int): amount of money in the currency of the source country name <i>toCountryName</i> (str): name of the target country
Output	(float): exchanged amount of money in the currency of the target country

```

class CurrencyRate(object):
    def __init__(self):
        self.dates = []
        self.countries = []
        self.exchangeRates = []

    def insertDate(self, date):
        self.dates.append(date)

    def insertCountry(self, country):
        self.countries.append(country)

    def insertExchangeRate(self, rate):
        self.exchangeRates.append(rate)

    def change(self, dateInt, fromCountryName, fromMoneyAmount, toCountryName):
        dateIndex = -1
        fromCountryIndex = -1
        toCountryIndex = -1
        for i in range(len(self.dates)):
            if self.dates[i] == dateInt:
                dateIndex = i
                break
        for i in range(len(self.countries)):
            if fromCountryName == self.countries[i][0]:
                fromCountryIndex = i
            if toCountryName == self.countries[i][0]:
                toCountryIndex = i

        return

```

4-3

return

4-4. (5 points) The following is the main procedure of the currency management program. Fill in the blank in order to set exchange rate information (e.g., ['AUSTRALIA', 0.8718, 0.8726, 0.8623, 0.861, 0.8674]) as a part of the attribute, 'exchangeRates' in the 'CurrencyRate' object.

Note that the line that includes date information starts with '<th id="a'.

The line that includes a country name starts with '<th id="r1" headers="a1">' tag.

The line that includes the currency of a country starts with '<td headers="a2 a1 r1">' tag. The line that includes a currency exchange rate starts with '<td headers="a◆', where ◆ changes from 3 to 7.

```
fin = open("weekly_exchange_rates.htm", "r")
currencyInfo = CurrencyRate()
j = 0
country = ""
countryInfo = []
for line in fin:
    if '<th id="a' in line:
        if not 'COUNTRY' in line and not 'CURRENCY' in line:
            currencyInfo.insertDate(Date(line))          # date
    elif '<th id="r1" headers="a1">' in line:              # Country name
        if not len(countryInfo) == 0:
            currencyInfo.insertCountry(countryInfo[:2])
            currencyInfo.insertExchangeRate(  )
            countryInfo = []
        countryInfo.append(parse(line).strip('*'))
        j = 3
    elif '<td headers="a2 a1 r1">' in line:                # Currency in the country
        countryInfo.append(parse(line).strip())
    elif str('<td headers="a%d a1 r1">' % j) in line:      # Exchange rates
        countryInfo.append(float(parse(line).strip()))
        j += 1
fin.close()
```