Python Programming

Lists processing II

Mariusz Dzieńkowski

Institute of Computer Science
Lublin University of Technology
m.dzienkowski@pollub.pl

List comprehensions

- List comprehensions provide a concise way to create a sequential list of elements.
- A list comprehension consists of brackets containing an expression followed by a for clause, then zero or more for or if clauses.

```
>>> list1 = [x for x in range(5)]
>>> list1
[0, 1, 2, 3, 4]
>>> list2 = [0.5*x for x in list1]
>>> list2
[0.0, 0.5, 1.0, 1.5, 2.0]
>>> list3 = [x for in list2 if x < 1.5]
>>> list3
[0.0, 0.5, 1.0]
```

List methods

Function	*Description*
`append(x: object): None`	Adds an element x to the end of the list.
`count(x: object): int`	Returns the number of items element x appears in the list.
`extend(l :list): None`	Appends all the elements in l to the list.
`index(x: object): int`	Returns the index of the first occurrence of element x in the list.
`insert(index: int, x: object): None`	Inserts an element x at a given index.
`pop(i): object`	Removes the element at the given position and returns it.
`remove(x: object): None`	Removes the first occurrence of element x from the list.
`reverse(): None`	Reverses the elements in the list.
`sort(): None`	Sorts the elements in the list in ascending order.

Splitting a string into a list

- The str class conatins the `split` method, which is useful for splitting items in a string into a list.

```
>>> items = ”Jane John Peter Susan”.split()
['Jane', 'John', 'Peter', 'Susan']
>>> items = ”04-12-2001”.split(”-”)
['04', '12', '2001']
```

Problem 1

Lists processing – an application that performs the following things:

- Initialization of the studentsDB list
- Function that returns two values:
  - a list with average grades of each student
  - maximum average for all students
- Function that shows a list of students
- Function that displays a list of students from a selected country
- Function that displays the best students

- Use the following function headers:

```
def maxAvg(sdb):
def show(sdb,avglst):
def selectCountry(sdb,country):
def bestStudents(sdb,avglst,max):
def main():
```

Problem 1 - solution

- Source code

```
studentsDB=[[1,"Ali", "Turkey",[3,3.5,4,5,5]],
            [2,"Claudio", "Italy",[2,4.5,3,2,1]],
            [3,"Jose", "Portugal",[3,4,5,5,4]],
            [4,"Mustafa","Turkey",[4,4,5,4,3]],
            [5,"Maz", "Morocco",[5,4,3,4,5]],
           ]
```

```
def maxAvg(sdb):
    avgList=[]
    for i in range(len(sdb)):
        sum=0
        for j in range(5):
            sum += sdb[i][3][j]
        avg = sum / 5
        avgList.append(avg)

    m=avgList[0]
    for i in range(1,len(avgList)):
        if m<avgList[i]:
            m = avgList[i]
    return avgList,m
```

Problem 1 - solution

- Source code

```
def show(sdb,avglst):
    print("\nList of students:")
    for i in range(len(sdb)):
        print(format(sdb[i][1],'10s'),format(avglst[i],'.2f'))

def selectCountry(sdb,country):
    x=1
    for i in sdb:
        if i[2]==country:
            print(x,'. ',i[1],sep='')
            x += 1

def bestStudents(sdb,avglst,max):
    print("The best students:")
    for i in range(0,len(avglst)):
        if avglst[i]==max:
            print(format(sdb[i][1],'10s'),format(avglst[i],'.2f'))
```

Problem 1 - solution

- Source code

```
def main():
    avgs,max=maxAvg(studentsDB)
    bestStudents(studentsDB,avgs,max)

    show(studentsDB,avgs)
    print("\nStudens from Turkey:")
    selectCountry(studentsDB,"Turkey")

main()
```

Problem 2

Checking the dimensions of luggage – a program that checks the size of luggage.

- Guidelines:
  - Define a function that:
    - gets data as a string parameter in a format like: 13x34x54
    - creates a list with 3 numbers as a dimensions: [width, height, depth]
    - calculates and returns a size of luggage
  - Define a main function that:
    - enters dimensions of luggage in a format like: 13x34x54
    - checks whether the size of luggage is bigger then 158 cm.

Problem 2 - solution

- Source code

```
def luggageDimensions(ld):
    numlist = [ int(i) for i in ld.split("x") ]
    product=1
    for i in numlist:
        product*=i
    return product

def main():
    x = input("Enter dimensions of luggage in a format e.g. 13x34x54")
    ldm = luggageDimensions(x)
    print('Dimensions of luggage =',ldm)
    if ldm > 158:
        print("You should pay extra money for your luggage.")
    else:
        print("You don't pay extra money for your luggage.")

main()
```

Problem 3

Magic square – a program that given a grid containing numbers, determines whether it's a magic square.

A magic square is an arrangement of distinct numbers (i.e., each number is used once), usually integers, in a square grid, where the numbers in each row, and in each column, and the numbers in the main and secondary diagonals, all add up to the same number, called the "magic constant ".

- Guidelines:
  - Verify magic squares of any size

    ```
    def isMagicSquare(ms,rows,columns):
    ```

  - Example inputs of magic squares:

```
array2 = [[8,1,6],
          [3,5,7],
          [4,9,2]]
```

```
array1 = [[23,6,19,2,15],
          [4,12,25,8,16],
          [10,18,1,14,22],
          [11,24,7,20,3],
          [17,5,13,21,9]]
```

Problem 3 - solution

- Source code

```
def isMagicSquare(ms,rows,columns):
    if rows!=columns:
        return False
    srl=[]
    for r in range(rows):
        s=0
        for c in range(columns):
            s += ms[r][c]
        srl.append(s)

    scl=[]
    for c in range(columns):
        s=0
        for r in range(rows):
            s += ms[r][c]
        scl.append(s)

    if srl!=scl:
        return False
    for i in range(rows):
        for j in range(columns):
            if srl[i]!=scl[j]:
                return False
```

```
    s = 0
    for i in range(rows):
        s += ms[i][i]

    if (s not in scl) and (s not in srl):
        return False

    s = 0
    for i in range(rows):
            s += ms[i][rows-1-i]

    if (s not in scl) and (s not in srl):
        return False
    return True
```

Problem 3 - solution

- Source code

```
def main():
    array1 = [[23,6,19,2,15],
              [4,12,25,8,16],
              [10,18,1,14,22],
              [11,24,7,20,3],
              [17,5,13,21,9]]
    array2 = [[8,1,6],
              [3,5,7],
              [4,9,2]]

    if isMagicSquare(array1,5,5):
        print('Square is magic.')
    else:
        print('Square is not magic.')

    if isMagicSquare(array2,3,3):
        print('Square is magic.')
    else:
        print('Square is not magic.')

main()
```