

Chapter 7. Lists and Tuples

Solutions for Programming Questions

Submitted by: 19251A0421

1. Valid Number Information

Program:

```
'''
Problem Statement:
Design a program that uses a loop to build a
list named valid_numbers that contains only
the numbers between 0 and 100 from the numbers list below.
The program should then determine
and display the total and average of the values
in the valid_numbers list.
numbers = [74, 19, 105, 20, -2, 67, 77, 124, -45, 38]
'''

import math
numbers=[74,19,105,20,-2,67,77,124,-45,30]
print("List of Numbers:",*numbers)
valid_numbers=[]      #output list
# Traverse 'numbers' list
# if any element is in between 0 to 100, \
# add that element into 'valid_numbers' list
for i in range(len(numbers)):
    if numbers[i]>=0 and numbers[i]<=100:
        valid_numbers.append(numbers[i])
#computing total of all valid numbers
total=sum(valid_numbers)
#computing average of all valid numbers
average=sum(valid_numbers)/len(valid_numbers)
#display output
print("List of valid numbers:",*valid_numbers)
print("Total of all valid numbers:",total)
print("Average of all valid numbers:",round(average,4))
```

Output:

```
List of Numbers: 74 19 105 20 -2 67 77 124 -45 30
List of valid numbers: 74 19 20 67 77 30
Total of all valid numbers: 287
Average of all valid numbers: 47.8333
```

2. Lottery Number Generator

Program:

```
'''
Problem Statement:
Design a program that generates a seven-digit lottery number.
The program should generate seven random numbers,
each in the range of 0 through 9,
and assign each number to a list element.
(Random numbers were discussed in Chapter 5.)
Then write another loop that displays the contents of the list.
'''

import random
#randint(m,n) will generate integers from m to n
lottery_number=[]
for i in range(7):
    ith_digit=random.randint(0,9)
    lottery_number.append(ith_digit)
print("Generated 7 digit Lotter Number is: ",*lottery_number)
```

Output:

```
Generated 7 digit Lotter Number is: 7726300
```

3. Rainfall Statistics

Program:

```
'''
Problem Statement:
Design a program that lets the user enter the total rainfall
for each of 12 months into a list.
The program should calculate and display
the total rainfall for the year,
the average monthly rainfall,
the months with the highest and lowest amounts.
'''

rainfall=[]
months=['January','February','March','April','May',\
        'June','July','August','September',\
        'October','November','December']

print("Enter rainfall in mms")
for i in range(12):
    print("Enter rainfall measure in",months[i],end=": ")
    level=int(input())
    rainfall.append(level)

#the total annual rainfall = sum of rainfall in all twelve
months
total=sum(rainfall)

#the average monthly rainfall = the total annual rainfall / 12
average=total/12

#the month with the highest rainfall
highest_rainfall=rainfall[0]
highest_rainfall_month=months[0]
for i in range(1,12):
    if rainfall[i]>highest_rainfall:
        highest_rainfall=rainfall[i]
        highest_rainfall_month=months[i]

#the month with lowest rainfall
lowest_rainfall=rainfall[0]
lowest_rainfall_month=months[0]
for i in range(1,12):
    if rainfall[i]<lowest_rainfall:
        lowest_rainfall=rainfall[i]
```

```
lowest_rainfall_month=months[0]

print("The total annual rainfall: ",total,"mm")
print("The average monthly rainfall: ",round(average,2),"mm")
print("The month with highest rainfall: "\
      ,highest_rainfall_month,"with",highest_rainfall,"mm")
print("The month with lowest rainfall: "\
      ,lowest_rainfall_month,"with",lowest_rainfall,"mm")
```

Output:

```
Enter rainfall in mms
Enter rainfall measure in January: 284
Enter rainfall measure in February: 172
Enter rainfall measure in March: 137
Enter rainfall measure in April: 240
Enter rainfall measure in May: 258
Enter rainfall measure in June: 212
Enter rainfall measure in July: 232
Enter rainfall measure in August: 230
Enter rainfall measure in September: 241
Enter rainfall measure in October: 311
Enter rainfall measure in November: 328
Enter rainfall measure in December: 332
The total annual rainfall:  2977 mm
The average monthly rainfall:  248.08 mm
The month with highest rainfall:  December with 332 mm
The month with lowest rainfall:  January with 137 mm
```

4.Number Analysis Program

Program:

```
'''
Problem Statement:
Design a program that asks the user to enter a series of 20
numbers.
The program should store the numbers in a list
then display the following data:
• The lowest number in the list
• The highest number in the list
• The total of the numbers in the list
• The average of the numbers in the list
'''

print("Enter Series of 20 Numbers: ")
arr=[]
for i in range(20):
    arr.append(int(input()))
lowest=min(arr)
highest=max(arr)
total=sum(arr)
average=total/20
print("the lowest number is: ",lowest)
print("the highest number is: ",highest)
print("the total of all numbers: ",total)
print("the average of all numbers: ",average)
```

Output:

```
Enter Series of 20 Numbers:
20
25
10
15
40
30
25
34
45
36
21
25
16
15
67
100
1
22
23
45
the lowest number is: 1
the highest number is: 100
the total of all numbers: 615
the average of all numbers: 51.25
□
```

5.Charge Account Validation

Program:

```
'''
Problem Statement
If you have downloaded the source code from the Premium
Companion Website you will find a file named
charge_accounts.txt in the Chapter 07 folder.
This file has a list of a company's valid charge account
numbers.
Each account number is a seven-digit number, such as 5658845.
Write a program that reads the contents of the file into a
list.
The program should then ask the user to enter a charge account
number.
The program should determine whether the number is
valid by searching for it in the list.
If the number is in the list, the program should
display a message indicating the number is valid.
If the number is not in the list,
the program should display a message indicating the number is
invalid.
(You can access the Premium Companion Website at
www.pearsonglobaleditions.com/gaddis.)
'''

file1=open("5. charge_accounts.txt",'r')
valid_charge_accounts=file1.read()
valid_charge_accounts=valid_charge_accounts.split('\n')
file1.close()

while(1):
    charge_account_input=input("Enter a charge account number:
")
    if charge_account_input.isnumeric() and
len(charge_account_input)==7:
        break
    else:
        print("Invalid Input")
        print("Please enter correct input")
if int(charge_account_input) in valid_charge_accounts:
    print("Account number is Valid")
else:
    print("Account number is Invalid")
```


Output:

```
Enter a charge account number: 4566778
Account number is Invalid
```

```
Enter a charge account number: 566789011
Invalid Input
Please enter correct input
Enter a charge account number: asdfghmj
Invalid Input
Please enter correct input
Enter a charge account number: 4fvv7800-
Invalid Input
Please enter correct input
Enter a charge account number: 5658845
Account number is Valid
```

6. Dice Rolling Function

Program:

```
'''
In a program, write a function named roll
that accepts an integer argument number_of_throws.
The function should generate and return
a sorted list of number_of_throws random numbers
between 1 and 6.
The program should prompt the user to enter a positive integer
that is sent to the function, and then print the returned list
'''

import random

def checkvalid(n):
    for i in range(len(n)):
        if n[i]<"0" or n[i]>"9":
            return 0
    return 1

def roll(no_of_throws):
    result=[]
    for i in range(no_of_throws):
        face_value=random.randint(1,6)
        result.append(face_value)
    result.sort()
    return result

number_of_throws=input("Enter no of throws you want to roll a
die (Note: Enter only positive integers): ")
while(True):
    if checkvalid(number_of_throws)==0:
        print("You have a entered a invalid number")
        print("Please enter a valid input")
        number_of_throws=input("Enter no of throws you want to
roll a die (Note: Enter only positive integers): ")
    else:
        number_of_throws=int(number_of_throws)
        break

rolled_values=roll(number_of_throws)
print(*rolled_values)
```

Output:

```
Enter no of throws you want to roll a die (Note: Enter only positive integers): -123
You have a entered a invalid number
Please enter a valid input
Enter no of throws you want to roll a die (Note: Enter only positive integers): 1.2
You have a entered a invalid number
Please enter a valid input
Enter no of throws you want to roll a die (Note: Enter only positive integers): -1.2
You have a entered a invalid number
Please enter a valid input
Enter no of throws you want to roll a die (Note: Enter only positive integers): 12
1 2 2 2 3 4 5 5 6 6 6 6
```

7. Driving License Exam

Program:

```
'''
The local driver's license office has asked you to create an
application
that grades the written portion of the driver's license exam.
The exam has 20 multiple-choice questions. Here
are the correct answers:
1. A
2. C
3. A
4. A
5. D
6. B
7. C
8. A
9. C
10. B
11. A
12. D
13. C
14. A
15. D
16. C
17. B
18. B
19. D
20. A
Your program should store these correct answers in a list.
The program should read the student's answers
for each of the 20 questions from a text file and
store the answers in another list.
(Create your own text file to test the application.)
After the student's answers
have been read from the file, the program should display a
message indicating whether the
student passed or failed the exam.
(A student must correctly answer 15 of the 20 questions to pass
the exam.)
It should then display the total number of correctly answered
questions,
the total number of incorrectly answered questions,
and a list showing the question numbers of the incorrectly
answered question
```

```

'''
file1=open("7. driving licence mcqs.txt",'r')
print(file1.read())
file1.close()

print("Enter your answers ")
print("Options lie in [A,B,C,D]")
student_answers=[]
key='ACAADBCACBADCADCBBD A'
wrongly_answeres_qns=[]
no_of_crct_answers=0
for i in range(20):
    while(1):
        print(i+1,end=". ")
        ans=input()
        if ans=='A' or ans=='B' or ans=='C' or ans=='D':
            break
        else:
            print("Enter valid option (Note:Correct Option is
one of A,B.C and D)")
            student_answers.append(ans)
            if ans==key[i]:
                no_of_crct_answers+=1
            else:
                wrongly_answeres_qns.append(i+1)
print("You have Scored",no_of_crct_answers,"marks")
if no_of_crct_answers>=15:
    print("You have passed this exam")
else:
    print("You have failed in this exam")
if no_of_crct_answers!=20:
    print("You answers are incorrect for following questions:
")
    print(*wrongly_answeres_qns)

```

Output:

- 1) The lap belt part of your seatbelt should rest where?
 - a) Across your hips
 - b) Just below your ribs
 - c) Across your stomach
 - d) Across your thighs

- 2) The first thing you should do before making a right turn is what?
 - a) Turn the wheel
 - b) Accelerate
 - c) Check your mirrors
 - d) Signal

- 3) You are approaching a roundabout at which you want to turn left. Which course should you follow?
 - a) Left hand lane all the way round
 - b) Right hand lane on entry, left on exit
 - c) Left hand lane on entry, right on exit
 - d) Right hand lane all the way round

- 4) A flashing red light indicates what?
 - a) Stop, then proceed cautiously
 - b) Maintain your speed and proceed cautiously
 - c) Speed up and proceed cautiously
 - d) Slow down and proceed cautiously

- 5) A fan belt should have about what amount of play?
 - a) 0.5 cm
 - b) 8 cm
 - c) 15 cm
 - d) 3 cm

- 6) If you feel an overtake is slightly risky or doubtful, what should you do?
- a) Try to make it but be prepared to pull out
 - b) Abandon the maneuver
 - c) Increase speed to make sure you make it
 - d) Sound your horn continuously as you make the maneuver
- 7) You are driving up a three lane one-way street and you wish to go straight ahead. Which lane should you choose?
- a) The right-hand lane
 - b) The left-hand lane
 - c) The middle lane
 - d) Whichever one you prefer
- 8) When going around the roundabout at which you wish to turn right, you should switch to indicating left...?
- a) After you have passed the exit before the one you want
 - b) At no time
 - c) Once you are half way around the roundabout
 - d) As soon as you enter the roundabout
- 9) You are approaching a wild animal on the road. What should you do?
- a) Accelerate and give them a wide berth
 - b) Sound your horn continually
 - c) Slow down and give them a wide berth
 - d) Rev your engine loudly
- 10) If your accelerator sticks when you are driving, how should you react?
- a) Sound your horn continuously
 - b) Shift gear into neutral
 - c) Stamp on the brakes as hard as possible
 - d) Pull on the parking brake

11) Which of these persons has a right to make you stop by signaling to you?

- a) All of them
- b) School crossing attendants
- c) Police officers
- d) Traffic wardens

12) Any four wheeled vehicle you drive on the road must have...?

- a) A full windscreen
- b) Rear seats
- c) A lockable trunk
- d) Registration plates front and rear

13) If you start to feel drowsy when driving, which of these should you do?

- a) Drive more slowly
- b) Increase your speed
- c) Pull over and take a nap
- d) Close the windows

14) Under what circumstances is it acceptable not to stop after an accident?

- a) None
- b) If you are in a hurry
- c) If nobody is hurt
- d) If it wasn't your fault

15) When parking, you should always be within what distance of the curb?

- a) 90 cm
- b) 70 cm
- c) 50 cm
- d) 30 cm

- 16) When driving on a four lane road with a dividing median, which position should you choose?
- a) The left-hand lane
 - b) Whichever one you like
 - c) The right-hand lane
 - d) Straddling both lanes
- 17) At a roundabout, you must give way to traffic...?
- a) That is turning left
 - b) Coming from the right
 - c) That is larger than you
 - d) Coming from the left
- 18) Before setting off on the road, which of these should you do?
- a) Shout a warning to others
 - b) Check all around and under your vehicle for children and animals
 - c) Move away as quickly as possible
 - d) Sound your horn to warn others you will be moving
- 19) If you are stressed, excited or upset, how should you approach driving?
- a) Take quieter streets
 - b) Drive more slowly than usual
 - c) Drive with the windows open
 - d) Avoid driving until you have calmed down
- 20) Your vehicle tires must have a tread depth of at least
- a) 1.60 mm.
 - b) 0.75 mm.
 - c) 0.5 mm.
 - d) 0.25 mm.

Ln:62 Col:50 Spaces:4 Lf5:8 CR15

Enter your answers

Options lie in [A,B,C,D]

1. a

Enter valid option (Note:Correct Option is one of A,B.C and D)

1. aa

Enter valid option (Note:Correct Option is one of A,B.C and D)

1. AA

Enter valid option (Note:Correct Option is one of A,B.C and D)

1. AB

Enter valid option (Note:Correct Option is one of A,B.C and D)

1. 0

Enter valid option (Note:Correct Option is one of A,B.C and D)

1. 1

Enter valid option (Note:Correct Option is one of A,B.C and D)

1. .

Enter valid option (Note:Correct Option is one of A,B.C and D)

1. ,

Enter valid option (Note:Correct Option is one of A,B.C and D)

1. A

2. B

3. C

4. D

5. E

2. B

3. C

4. D

5. E

Enter valid option (Note:Correct Option is one of A,B.C and D)

5. F

Enter valid option (Note:Correct Option is one of A,B.C and D)

5. G

Enter valid option (Note:Correct Option is one of A,B.C and D)

5. A

6. D

7. C

8. B

9. B

10. B

11. B

12. B

13. B

14. B

15. B

16.

Enter valid option (Note:Correct Option is one of A,B.C and D)

16. B

17. B

18. B

19. B

20. B

You have Scored 5 marks

You have failed in this exam

You answers are incorrect for following questions:

2 3 4 5 6 8 9 11 12 13 14 15 16 19 20

8. Name Search

Program:

```
'''
If you have downloaded the source code
you will find the following files in the Chapter 07 folder:
•   GirlNames.txt This file contains a list
of the 200 most popular names given to girls
born in the United States from the year 2000 through 2009.
•   BoyNames.txt This file contains a list of the 200 most
popular names given to boys
born in the United States from the year 2000 through 2009
Write a program that reads the contents of the two files into
two separate lists.
The user should be able to enter a boy's name, a girl's name,
or both, and the application will display
messages indicating whether the names were among the most
popular.
(You can access the Premium Companion Website at
www.pearsonglobaleditions.com/gaddis.
'''

girl_names_file=open("8. GirlNames.txt",'r')
girls_list=girl_names_file.read()
girls_list=girls_list.split("\n")
girl_names_file.close()

boy_names_file=open("8. BoyNames.txt",'r')
boys_list=boy_names_file.read()
boys_list=boys_list.split("\n")
boy_names_file.close()

print("Enter list of names: ")
names=list(input().split())
for i in range(len(names)):
    names[i]=names[i].title() #capitalizes each word
    if names[i] in girls_list:
        print(names[i],"is one of the most popular girl's
name")
    elif names[i] in boys_list:
        print(names[i],"is one of the most popular boy's name")
    else:
        print(names[i],"is not among most popular names of
girls and boys")
```

Output:

```
Enter list of names:
bradley jalen Hilbert Anu Kylee virat abby xavier juliat
Bradley is one of the most popular boy's name
Jalen is one of the most popular boy's name
Hilbert is not among most popular names of girls and boys
Anu is not among most popular names of girls and boys
Kylee is one of the most popular girl's name
Virat is not among most popular names of girls and boys
Abby is one of the most popular girl's name
Xavier is one of the most popular boy's name
Juliat is not among most popular names of girls and boys
```

9. Population Data

Program:

```
'''
    Population Data
    If you have downloaded the source code
    you will find a file named USPopulation.txt in the Chapter 07
    folder.
    The file contains the midyear population of the United States,
    in thousands, during the years 1950 through 1990.
    The first line in the file contains the population for 1950,
    the second line contains the population for 1951, and so
    forth.
    Write a program that reads the file's contents into a list.
    The program should display the following data:
    • The average annual change in population during the time
    period
    • The year with the greatest increase in population during the
    time period
    • The year with the smallest increase in population during the
    time period
'''

file1=open("9. USPopulation.txt",'r')
population_list=file1.read()
population_list=list(map(int,population_list.split('\n')))
file1.close()

start_year=1950
annual_changes_list=[]
for i in range(0,len(population_list)-1):
    annual_change=population_list[i+1]-population_list[i]
    annual_changes_list.append(annual_change)

average_change=sum(annual_changes_list)/len(annual_changes_list)
)
greatest_increase=max(annual_changes_list)
gi_index=annual_changes_list.index(greatest_increase)
smallest_increase=min(annual_changes_list)
si_index=annual_changes_list.index(smallest_increase)

print("The average annual change in population from 1950 to
1990 is",average_change)
print("The year with the greatest increase in population is
from",gi_index+1950,"to",gi_index+1951)
```

```
print("The year with the smallest increase in population is  
from",\  
      si_index+1950,"to",si_index+1951)
```

Output:

```
The average annual change in population from 1950 to 1990 is 2443.875  
The year with the greatest increase in population is from 1954 to 1955  
The year with the smallest increase in population is from 1966 to 1967
```

10. World Series Champions

Program:

```
'''  
If you have downloaded the source code  
you will find a file named WorldSeriesWinners.txt in the  
Chapter 07 folder.  
This file contains a chronological list of the  
World Series winning teams from 1903 through 2009.  
(The first line in the file is the name of the team that  
won in 1903, and the last line is the name of the team that won  
in 2009.  
Note the World Series was not played in 1904 or 1994.)  
Write a program that lets the user enter the name of a team,  
then displays the number of  
times that team has won the World Series in the time period  
from 1903 through 2009.  
TIP: Read the contents of the WorldSeriesWinners.txt file into  
a list. When the  
user enters the name of a team, the program should step through  
the list, counting the  
number of times the selected team appears  
'''  
  
file1=open("10. WorldSeriesWinners.txt",'r')  
winners_list=file1.read()  
winners_list=list(winners_list.split('\n'))  
file1.close()  
  
if no_of_times>0:  
    years=[]  
    j=0  
    for i in range(1903,2009):  
        if i==1904 or i==1994:  
            continue  
        else:
```

```
        if name_of_team==winners_list[j]:
            years.append(i)
            j=j+1
        print(name_of_team,"won",no_of_times,"times","in
years",end=" ")
        print(*years)
    else:
        print(name_of_team,"has never won match / Invalid Input")
```

Output:

```
Enter the name of team: Boston Red Sox
Boston Red Sox won 6 times in years 1912 1915 1916 1918 2004 2007
```

```
Enter the name of team: Mumbai Indians
Mumbai Indians has never won match / Invalid Input
```

11. Lo Shu Magic Square

Program:

```
'''
The Lo Shu Magic Square is a grid with 3 rows and 3 columns,
shown in Figure 7-18.
    The Lo Shu Magic Square has the following properties:
    • The grid contains the numbers 1 through 9 exactly.
    • The sum of each row, each column, and each diagonal all
      add up to the same number.
This is shown in Figure 7-19.
In a program you can simulate a magic square using a two-
dimensional list.
Write a function that accepts a two-dimensional list as an
argument and
determines whether the list is a Lo Shu Magic Square.
    Test the function in a program
'''

def magicSquare(grid):
    '''
        a00      a01      a02
        a10      a11      a12
        a20      a21      a22
    '''
    r0=sum(grid[0])
    r1=sum(grid[1])
    r2=sum(grid[2])
    dij=grid[0][0]+grid[1][1]+grid[2][2]
    din_j=grid[2][0]+grid[1][1]+grid[0][2]
    c0=grid[0][0]+grid[1][0]+grid[2][0]
    c1=grid[0][1]+grid[1][1]+grid[2][1]
    c2=grid[0][2]+grid[1][2]+grid[2][2]
    if r0==r1==r2==dij==din_j==c0==c1==c2:
        return 1
    else:
        return 0

print("Enter elements in 3-D Grid: ")
print("Note: Enter numbers only between 1 and 9")
grid=[]
for i in range(3):
    row=list(map(int,input().split()))
    grid.append(row)
```



```
flag=magicSquare(grid)
if flag==1:
    print("Given 3x3 grid is a Lo Shu Magic Square")
else:
    print("Given 3x3 grid is not a Lo Shu Magic Square")
```

Output:

```
Enter elements in 3-D Grid:
Note: Enter numbers only between 1 and 9
4 9 2
3 5 7
8 1 6
Given 3x3 grid is a Lo Shu Magic Square
```

```
Enter elements in 3-D Grid:
Note: Enter numbers only between 1 and 9
4 8 2
3 5 7
8 1 6
Given 3x3 grid is not a Lo Shu Magic Square
```

```
Enter elements in 3-D Grid:
Note: Enter numbers only between 1 and 9
4 9 2
3 5 7
8 1 6
Given 3x3 grid is a Lo Shu Magic Square
```