

# [Solution] CP02

July 3, 2022

## 1 Class Partipation 02

### 1.0.1 01. Budget Analysis

Write a program that asks the user to enter the amount that he or she has budgeted for a month. A loop should then prompt the user to enter each of his or her expenses for the month and keep a running total. When the loop finishes, the program should display the amount that the user is over or under budget.

```
[1]: budget = float(input('How much have you budgeted this month? '))
expense = 0
check = 0
while check >= 0:
    check = float(input('Enter an expense amount or -1 to quit: '))
    if check != -1:
        expense += check
        balance = budget - expense
if balance < 0:
    print('You have not budgeted enough, You are going to be short this month')
elif balance == 0:
    print('Be careful, You have budgeted enough to make it through this month')
else:
    print('You will have extra {} this month'.format(balance))
```

```
How much have you budgeted this month? 10000
Enter an expense amount or -1 to quit: 1000
Enter an expense amount or -1 to quit: 800
Enter an expense amount or -1 to quit: 500
Enter an expense amount or -1 to quit: -1
You will have extra 7700.0 this month
```

### 1.0.2 02. Average Rainfall

Write a program that uses nested loops to collect data and calculate the average rainfall over a period of years. The program should first ask for the number of years. The outer loop will iterate once for each year. The inner loop will iterate twelve times, once for each month. Each iteration of the inner loop will ask the user for the inches of rainfall for that month. After all iterations, the program should display the number of months, the total inches of rainfall, and the average rainfall per month for the entire period.

```
[2]: num_year = int(input('Enter number of years: '))
rainfall = 0
for year in range(1, num_year + 1):
    for month in range(1, 13):
        rain_value = float(input('Enter no of inches of rainfall for month {}_
↳of year {} : '.format(month, year)))
    rainfall += rain_value
num_months = num_year * 12
avg_rainfall = rainfall / num_months
print('The total no of rainfall during these {} months was {} inches'.
↳format(num_months, rainfall))
print('The average rainfall per month was ', round(avg_rainfall, 2))
```

```
Enter number of years: 1
Enter no of inches of rainfall for month 1 of year 1 : 5
Enter no of inches of rainfall for month 2 of year 1 : 5.1
Enter no of inches of rainfall for month 3 of year 1 : 5.5
Enter no of inches of rainfall for month 4 of year 1 : 6
Enter no of inches of rainfall for month 5 of year 1 : 4.859
Enter no of inches of rainfall for month 6 of year 1 : 6.4
Enter no of inches of rainfall for month 7 of year 1 : 4.2
Enter no of inches of rainfall for month 8 of year 1 : 9
Enter no of inches of rainfall for month 9 of year 1 : 4
Enter no of inches of rainfall for month 10 of year 1 : 3
Enter no of inches of rainfall for month 11 of year 1 : 6.2
Enter no of inches of rainfall for month 12 of year 1 : 5.5547
The total no of rainfall during these 12 months was 5.5547 inches
The average rainfall per month was 0.46
```

### 1.0.3 03. Ocean Level

Assuming the ocean's level is currently rising at about 1.6 millimeters per year, create an application that displays the number of millimeters that the ocean will have risen each year for the next 25 years.

```
[3]: print('The number of millimeters that the ocean will have risen each year for_
↳the next 25 years.')
for year in range(2022, 2021 + 26):
    ocean_level = (year - 2020) * 1.6
    print(format(year, '<15d'), format(ocean_level, ',.1f'))
```

The number of millimeters that the ocean will have risen each year for the next 25 years.

2022	3.2
2023	4.8
2024	6.4
2025	8.0
2026	9.6

2027	11.2
2028	12.8
2029	14.4
2030	16.0
2031	17.6
2032	19.2
2033	20.8
2034	22.4
2035	24.0
2036	25.6
2037	27.2
2038	28.8
2039	30.4
2040	32.0
2041	33.6
2042	35.2
2043	36.8
2044	38.4
2045	40.0
2046	41.6

#### 1.0.4 04. Rainfall Statistics

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, and the months with the highest and lowest amounts.

```
[4]: MONTHS = ['January', 'Feburary', 'March', 'April', 'May', 'June', 'July',
    ↪ 'August', 'September', 'October', 'November',
    'December']
```

```
# This function calculates the sum of list of numbers
```

```
def total(List):
```

```
    accum = 0
```

```
    for e in List:
```

```
        accum += e
```

```
    return accum
```

```
# This function calculates the average of list of numbers
```

```
def average(List):
```

```
    return total(List) / len(List)
```

```
# This list contains the amount of rainfall of each month
```

```
monthly_rain = []
```

```

# Here we allow the user to fill in the list monthly_rain
for month in MONTHS:
    monthly_rain.append(float(input('Enter total rainfall for month {} in cm: '.
    ↪format(month))))

# All variable we need to display
total_rain = total(monthly_rain)
average_rain = average(monthly_rain)
max_rain = max(monthly_rain)
month_of_max_rain = MONTHS[monthly_rain.index(max(monthly_rain))]
min_rain = min(monthly_rain)
month_of_min_rain = MONTHS[monthly_rain.index(min(monthly_rain))]

# Displaying all the rainfall info
print('Total rainfall this year was {} in cm'.format(total_rain))
print('The average monthly rainfall was {} in cm'.format(average_rain))
print('The max rainfall was {} in cm'.format(max_rain))
print('The month of max rainfall was {} in cm'.format(month_of_max_rain))
print('The min rainfall was {} in cm'.format(min_rain))
print('The month of min rainfall was {} in cm'.format(month_of_min_rain))

```

```

Enter total rainfall for month January in cm: 12
Enter total rainfall for month Feburary in cm: 14
Enter total rainfall for month March in cm: 15
Enter total rainfall for month April in cm: 10
Enter total rainfall for month May in cm: 9
Enter total rainfall for month June in cm: 10.14505
Enter total rainfall for month July in cm: 7.5
Enter total rainfall for month August in cm: 4.235
Enter total rainfall for month September in cm: 5.2
Enter total rainfall for month October in cm: 4.35
Enter total rainfall for month November in cm: 11
Enter total rainfall for month December in cm: 1.58
Total rainfall this year was 104.01004999999999 in cm
The average monthly rainfall was 8.667504166666665 in cm
The max rainfall was 15.0 in cm
The month of max rainfall was March in cm
The min rainfall was 1.58 in cm
The month of min rainfall was December in cm

```

### 1.0.5 05. Lo Shu Magic Square

The Lo Shu Magic Square is a grid with 3 rows and 3 columns. The Lo Shu Magic Square has the following properties:

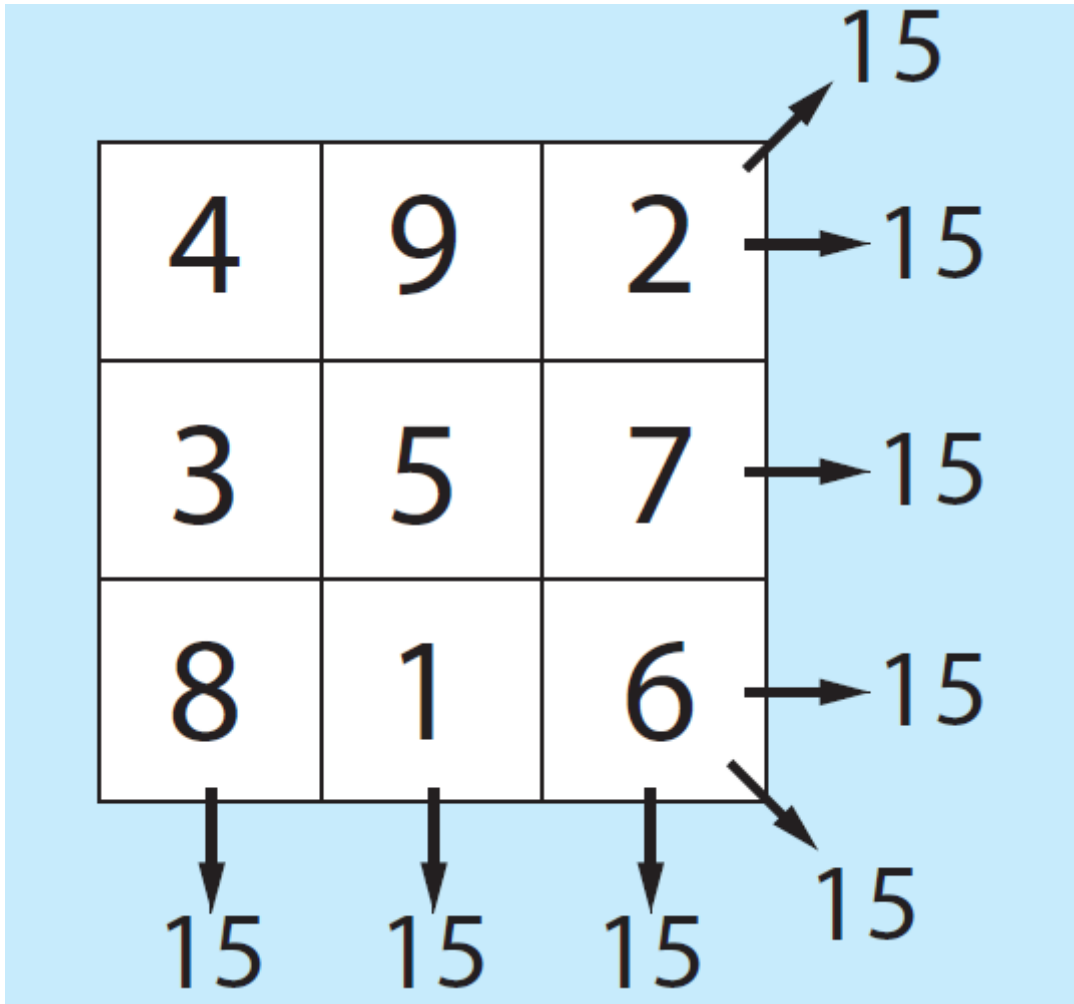
- The grid contains the numbers 1 through 9 exactly.
- The sum of each row, each column, and each diagonal all add up to the same number.

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.

4	9	2
3	5	7
8	1	6

---



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

def row_sum(grid, row_no):
    sum = 0
    for row in range(row_no, row_no + 1):
        for col in range(len(grid[row])):
            sum += grid[row][col]
    return sum

def col_sum(grid, col_no):
    sum = 0
    for row in range(len(grid)):
        for col in range(col_no, col_no + 1):
            sum += grid[row][col]
    return sum
```

```

if row_sum(grid, 0) == row_sum(grid, 1) and row_sum(grid, 1) == row_sum(grid, 2) and col_sum(grid, 0) == col_sum(grid, 1) and col_sum(grid, 1) == col_sum(grid, 2):
    print('There is a magic square')
else:
    print('Given matrix is not a magic square')

```

There is a magic square