Python Basic Exercise #6

Total Question: **7 Question**
Topic: **Problem Solving**

1. Create a program that can find the largest value and the smallest value in a list. [41, 5, 1, 3, 89, 32]

Hint:

- Use a python list
- Create 2 variables to store the min and max values
- Use python conditional statements.
- Use python looping.

2. Create a program that can *sort* data in a list, from smallest to largest, WITHOUT using the sort method found in Python List. [41, 5, 1, 3, 89, 32]

Hint:

- Use a python list
- Use python conditional statements.
- Use python looping.

3. "Student's Grade Mini Program"

Please follow the instruction.

- Create a Dictionary:
  - Create a dictionary named `student_grades` that stores student names as keys and their grades as values.
  - Add the following students and their grades:
    - Alice: 85
    - Bob: 92
    - Charlie: 78
- Update the Dictionary:
  - Add a new student, `David` , with a grade of 88.
  - Update Alice's grade to 90.
- Retrieve Data:
  - Print Bob's grade.
  - Check if Eve is in the dictionary and print an appropriate message.
- Delete a Key-Value Pair:
  - Remove Charlie from the dictionary.
- Loop through the Dictionary:
  - Print all student names and their grades in the format: "**Student: [name], Grade: [grade]**"

4. "Student's Attendance Mini Program"

Please follow the instruction.

- Choose the data structure below:
  - Nested Disctionary
  - List of Dictionary
  - Nested list
  - Dictionary of list
- Create a `school_data` that stores student information.
  - Add the following students with their grades and attendance:
    - Alice: grades: [85, 90, 88], attendance: 95
    - Bob: grades: [92, 88, 79], attendance: 88
    - Charlie: grades: [78, 85, 80], attendance: 99
    - David: grades: [88, 76, 90], attendance: 85
- Calculate Average Grades:
  - For each student, calculate the average of their grades and store it in their data.
- Update Attendance:
  - Increase the attendance of each student by 2% (attendance should not exceed 100%).
- Add a New Student:
  - Add a new student, Eve, with grades: [91, 85, 87] and attendance: 90.
- Retrieve and Print Data:
  - Print each student's name, average grade, and updated attendance in a formatted manner.
- Delete Students with Low Attendance:
  - Remove any student from the data who has an attendance less than 90%.

Expected output:

```
Student: Alice, Average Grade: 87.67, Attendance: 97%
Student: Bob, Average Grade: 86.33, Attendance: 90%
Student: Charlie, Average Grade: 81.00, Attendance: 94%
Student: David, Average Grade: 84.67, Attendance: 87%
Student: Eve, Average Grade: 87.67, Attendance: 90%

After removing students with low attendance:
Student: Alice, Average Grade: 87.67, Attendance: 97%
Student: Bob, Average Grade: 86.33, Attendance: 90%
Student: Charlie, Average Grade: 81.00, Attendance: 94%
Student: Eve, Average Grade: 87.67, Attendance: 90%
```

5. "Market Program"

- Use the "market.py" file created in the previous exercise.
- Create a CRUD feature.

Menu 1. Menampilkan Daftar Buah & Menu 5. Exit Program

```
Selamat Datang di Pasar Buah

List Menu :
1. Menampilkan Daftar Buah
2. Menambah Buah
3. Menghapus Buah
4. Membeli Buah
5. Exit Program
Masukkan angka Menu yang ingin dijalankan : 1
Daftar Buah

Index   | Nama    | Stock   |  Harga
0       | Apel    | 20      |  10000
1       | Jeruk   | 15      |  15000
2       | Anggur  | 25      |  20000

Selamat Datang di Pasar Buah

List Menu :
1. Menampilkan Daftar Buah
2. Menambah Buah
3. Menghapus Buah
4. Membeli Buah
5. Exit Program
Masukkan angka Menu yang ingin dijalankan : 5
```

Menu 2. Menambah Buah

```
Selamat Datang di Pasar Buah

List Menu :
1. Menampilkan Daftar Buah
2. Menambah Buah
3. Menghapus Buah
4. Membeli Buah
5. Exit Program

Masukkan angka Menu yang ingin dijalankan : 2
Masukkan Nama Buah      : Nanas
Masukkan Stock Buah     : 10
Masukkan Harga Buah     : 25000
Daftar Buah

Index   | Nama    | Stock   |  Harga
0       | Apel    | 20      |  10000
1       | Jeruk   | 15      |  15000
2       | Anggur  | 25      |  20000
3       | Nanas   | 10      |  25000

Selamat Datang di Pasar Buah

List Menu :
1. Menampilkan Daftar Buah
2. Menambah Buah
3. Menghapus Buah
4. Membeli Buah
5. Exit Program

Masukkan angka Menu yang ingin dijalankan :
```

Menu 3. Menghapus Buah

```
Selamat Datang di Pasar Buah

List Menu :
1. Menampilkan Daftar Buah
2. Menambah Buah
3. Menghapus Buah
4. Membeli Buah
5. Exit Program

Masukkan angka Menu yang ingin dijalankan : 3
Daftar Buah

Index   | Nama    | Stock   |  Harga
0       | Apel    | 20      |  10000
1       | Jeruk   | 15      |  15000
2       | Anggur  | 25      |  20000
Masukkan index buah yang ingin dihapus : 1
Daftar Buah

Index   | Nama    | Stock   |  Harga
0       | Apel    | 20      |  10000
1       | Anggur  | 25      |  20000

Selamat Datang di Pasar Buah

List Menu :
1. Menampilkan Daftar Buah
2. Menambah Buah
3. Menghapus Buah
4. Membeli Buah
5. Exit Program

Masukkan angka Menu yang ingin dijalankan :
```

Menu 4. Membeli Buah

```
List Menu :
1. Menampilkan Daftar Buah
2. Menambah Buah
3. Menghapus Buah
4. Membeli Buah
5. Exit Program

Masukkan angka Menu yang ingin dijalankan : 4
Daftar Buah

Index | Nama | Stock | Harga
0 | Apel | 20 | 10000
1 | Jeruk | 15 | 15000
2 | Anggur | 25 | 20000
Masukkan index buah yang ingin dibeli : 1
Masukkan jumlah yang ingin dibeli : 16
Stock tidak cukup, stock Jeruk tinggal 15
Isi Cart :
Nama | Qty | Harga
Mau beli yang lain? (ya/tidak) = ya
Masukkan index buah yang ingin dibeli : 1
Masukkan jumlah yang ingin dibeli : 5
Isi Cart :
Nama | Qty | Harga
Jeruk | 5 | 15000
Mau beli yang lain? (ya/tidak) = ya
Masukkan index buah yang ingin dibeli : 0
Masukkan jumlah yang ingin dibeli : 4
Isi Cart :
Nama | Qty | Harga
Jeruk | 5 | 15000
Apel | 4 | 10000
Mau beli yang lain? (ya/tidak) = tidak
Daftar Belanja :
Nama | Qty | Harga | Total Harga
Jeruk | 5 | 15000 | 75000
Apel | 4 | 10000 | 40000
Total Yang Harus Dibayar = 115000
Masukkan jumlah uang : 125000
Terima kasih

Uang kembali anda : 10000
```