WEEK-1

Design a Data structure for handling Student records- Design a solution, Implementation (Using Basic DS).

Data Structure

A data structure is a way of organizing data in computer memory, implemented in a programming language. This organization is required for efficient storage, retrieval, and modification of data.

Types of data structure

* Lists

Lists in Python are implemented as dynamic mutable arrays which hold an ordered collection of items.

### Sets

Sets in Python can be defined as mutable dynamic collections of **immutable unique** elements.

### Tuples

Tuples are almost identical to lists, so they contain an ordered collection of elements, except for one property: they are **immutable**.

### Dictionaries

Dictionaries in Python are very similar to real-world dictionaries. These are **mutable** data structures that contain a collection of **keys** and, associated with them, **values**. This structure makes them very similar to word-definition dictionaries.

Program:



Fig1.1:Screenshot of a program using data structures

Output

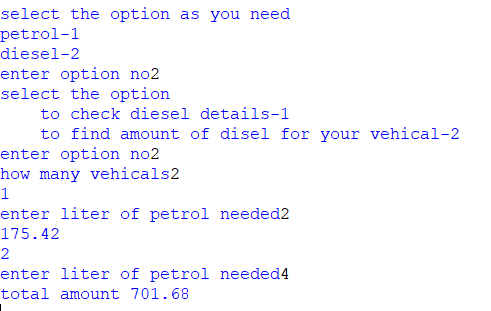


Fig:Screenshot of the output of above program

**WEEK-2**

Design a data structure for handling student records- Designing a solution, Implementation (Using ADT)

ADT(Abstract Data Type)

The abstract datatype is special kind of datatype, whose behavior is defined by a set of values and set of operations. The keyword “Abstract” is used as we can use these datatypes, we can perform different operations

advantages to using ADTs:

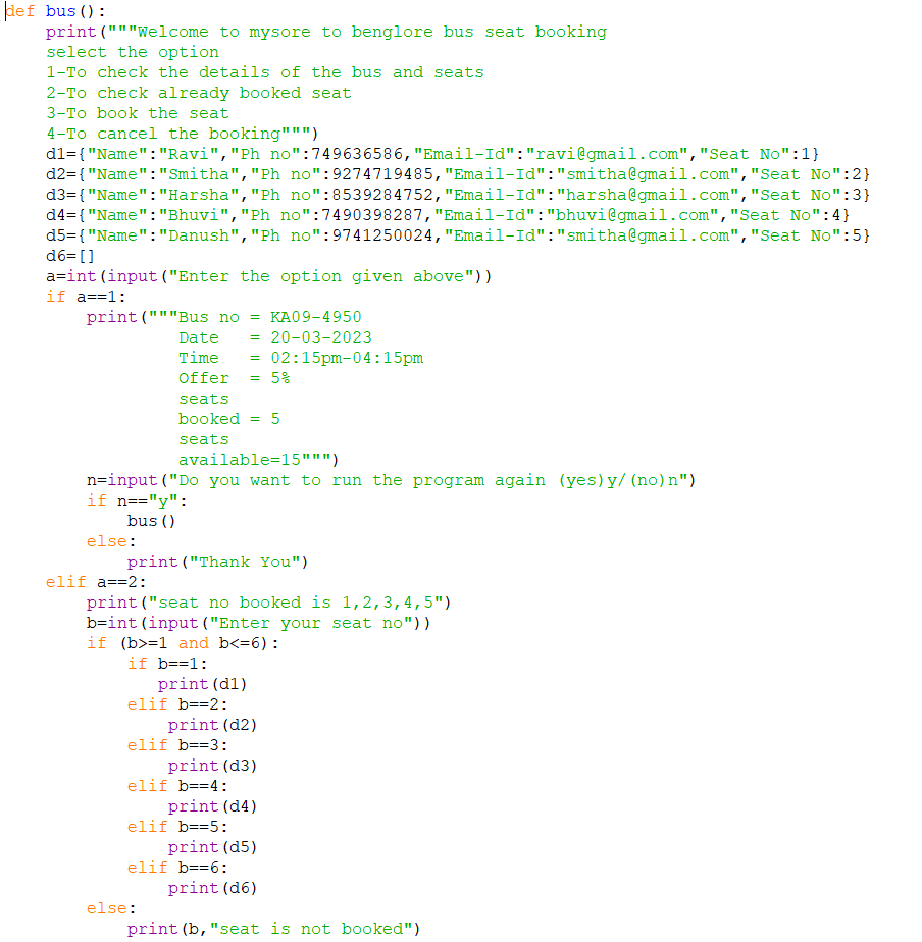
* ADTs are data representation independent.
* ADTs enable code reuse.
* ADTs enable clean extraction.

But there are also drawbacks:

* ADTs do not support fix/match syntax.
* ADTs have to expose derived operations.
* ADTs prohibit computational reasoning.

.

Program:

Fig2.6:Screenshot of a program using data structures

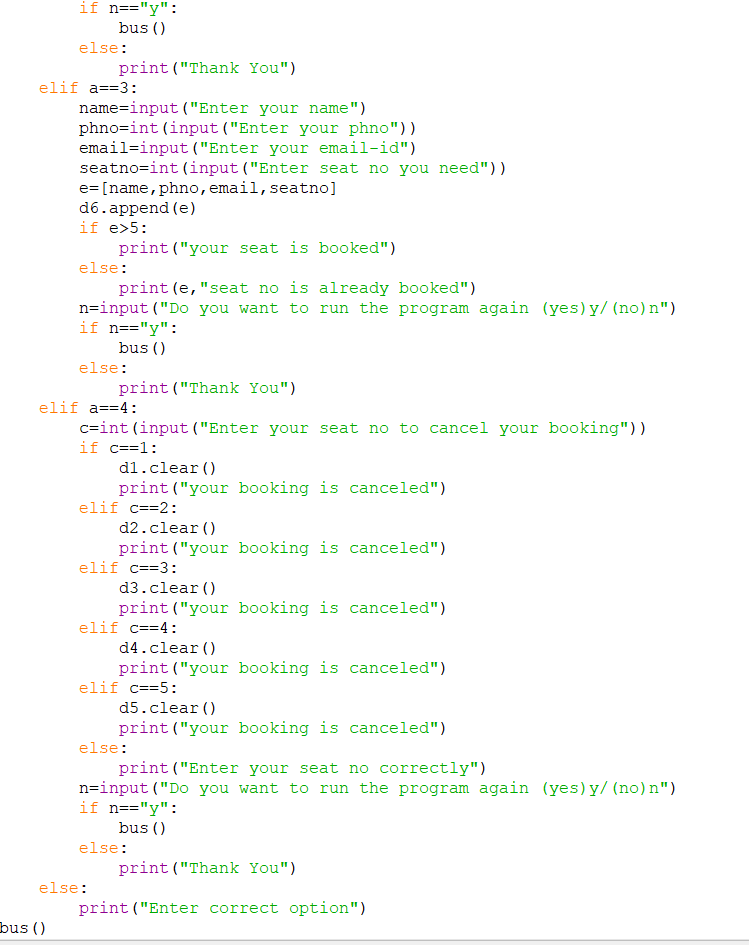


Fig2.7:Screenshot of a program using data structures

Output:

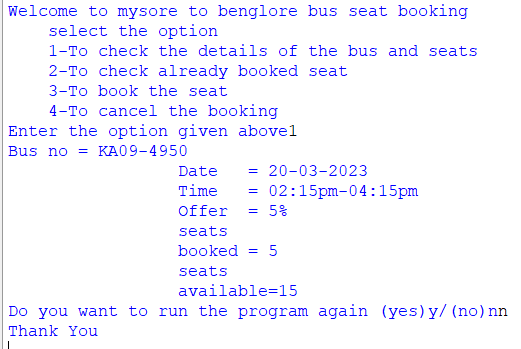


Fig2.8:Screenshot of the output of above program

WEEK-3

Optimize your solution (Bubble sort, selection sort, Insertion sort)

Sorting;

The arrangement of data in a preferred order is called sorting in the data structure. By sorting data, it is easier to search through it quickly and easily. The simplest example of sorting is a dictionary

Types of sorting:

1. Bubble sort,
2. Selection sort
3. Insertion sort

**Bubble sort :**

* **Bubble sort** is [a sorting algorithm](https://www.programiz.com/dsa/sorting-algorithm) that compares two adjacent elements and swaps them until they are in the intended order.
* Just like the movement of air bubbles in the water that rise up to the surface, each element of the array move to the end in each iteration. Therefore, it is called a bubble sort.

Program:

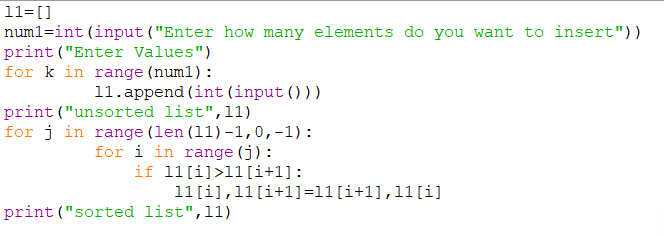


Fig3.1:Screenshot of a program using data structures

Output:

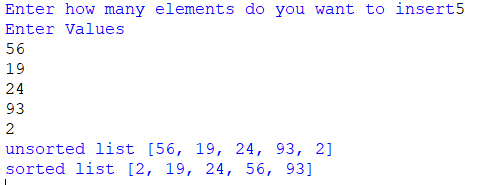


Fig3.2:Screenshot of the output of above program

Selection sort:

* Selection sort is an effective and efficient sort algorithm based on comparison operations.
* It adds one element in each iteration.
* You need to select the smallest element in the array and move it to the beginning of the array by swapping with the front element.
* You can also accomplish this by selecting the most potent element and positioning it at the back end.
* In each iteration, selection sort selects an element and places it in the appropriate position.

Program:

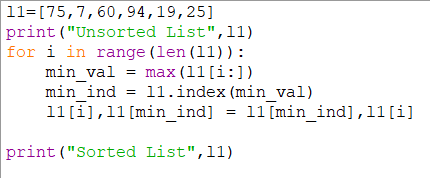


Fig3.3:Screenshot of a program using data structures

Output:

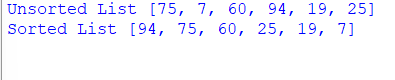


Fig3.4:Screenshot of the output of above program

Insertion sort:

* Insertion sort is [a sorting algorithm](https://www.programiz.com/dsa/sorting-algorithm) that places an unsorted element at its suitable place in each iteration.
* Insertion sort works similarly as we sort cards in our hand in a card game.
* We assume that the first card is already sorted then, we select an unsorted card. If the unsorted card is greater than the card in hand, it is placed on the right otherwise, to the left.

Program:

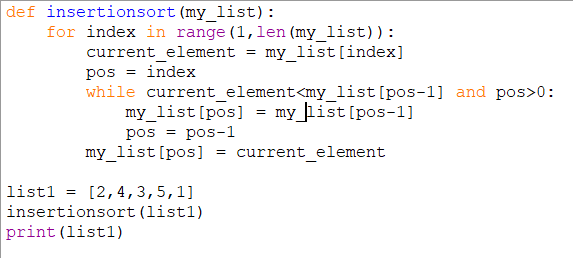


Fig3.5:Screenshot of a program using data structures

Output:



Fig3.6:Screenshot of the output of above program