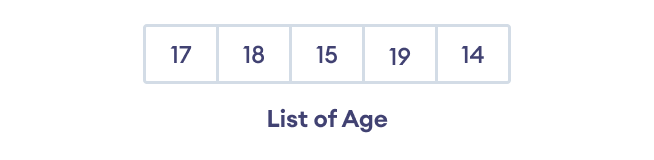
**Experiment Number – 6**

**Title – Write a program for Result Processing using nested list**

**Theory –**

A Python list is a collection of zero or more elements. A list in Python is used to store the sequence of various types of data. A list can be defined as a collection of values or items of different types. For example,

Suppose we need to record the ages of **5** students. Instead of creating **5** separate variables, we can simply create a list:



**Creating Python Lists**

A Python list is created by adding elements in the square brackets [ ].

For example:

num = [1, 2, 3, 4]

letter = [‘a’, ‘b’, ‘c’, ‘d’]

List items are ordered, changeable, and allow duplicate values. List items are indexed, the first item has index [0], the second item has index [1] etc. When we say that lists are ordered, it means that the items have a defined order, and that order will not change. If you add new items to a list, the new items will be placed at the end of the list. The list is changeable, meaning that we can change, add, and remove items in a list after it has been created. Since lists are indexed, lists can have items with the same value.

**Access Python List Elements**

In Python, each item in a list is associated with a number. The number is known as a list index. We can access elements of lists using the index number **(0, 1, 2 …)**. For example,

languages = ["Python", "Swift", "C++"]

# print item at index 0

print(languages[0])

## Negative Indexing in Python

Python allows negative indexing for its sequences. The index of **-1** refers to the last item, **-2** to the second last item and so on. For example,

languages = ["Python", "Swift", "C++"]

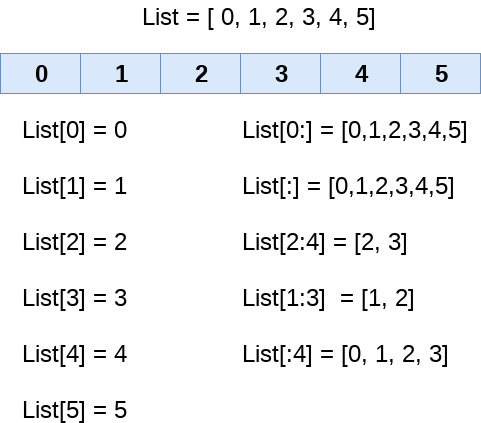
# access item at index 0

print(languages[-1])

## List Indexing and Splitting

The indexing is processed in the same way as it happens with the strings. The elements of the list can be accessed by using the slice operator [].

The index starts from 0 and goes to length - 1. The first element of the list is stored at the 0th index, the second element of the list is stored at the 1st index, and so on.



We can get the sub-list of the list using the following syntax.

list\_varible[start:stop:step]

* The **start** denotes the starting index position of the list.
* The **stop** denotes the last index position of the list.
* The **step** is used to skip the nth element within a **start:stop**

The initial index is represented by the start parameter, the ending index is determined by the step, and also the number of elements which are "stepped" through is the value of the end parameter. In the absence of a specific value for step, the default value equals 1. Inside the resultant SubList, the item also with index start would be present, but the one with the index finish will not. A list's initial element seems to have the index of 0.

**Python Basic List Operations**

The concatenation (+) and repetition (\*) operators work in the same way as they were working with the strings. The different operations of list are

1. Repetition - The repetition operator enables the list elements to be repeated multiple times.

lst = [12, 14, 16, 18, 20]

# repetition operator \*

l = lst \* 2

print(lst)

1. Concatenation – It concatenates the list mentioned on either side of the operator.

list1 = [12, 14, 16, 18, 20]

list2 = [9, 10, 32, 54, 86]

# concatenation operator +

lst = list1 + list2

print(lst)

1. Length - It is used to get the length of the list

list1 = [12, 14, 16, 18, 20, 23, 27, 39, 40]

# finding length of the list

len(list1)

1. Iteration - The for loop is used to iterate over the list elements.

list1 = [12, 14, 16, 39, 40]

for i in list1:

print(i)

1. Membership - It returns true if a particular item exists in a particular list otherwise false.

list1 = [100, 200, 300, 400, 500]

print(100 in list1)

## Python List Methods

Pythonhas a lot oflist methods that allow us to work with lists.

|  |  |
| --- | --- |
| **Method** | **Description** |
| [append()](https://www.programiz.com/python-programming/methods/list/append) | add an item to the end of the list |
| [extend()](https://www.programiz.com/python-programming/methods/list/extend) | add items of lists and other iterables to the end of the list |
| [insert()](https://www.programiz.com/python-programming/methods/list/insert) | inserts an item at the specified index |
| [remove()](https://www.programiz.com/python-programming/methods/list/remove) | removes item present at the given index |
| [pop()](https://www.programiz.com/python-programming/methods/list/pop) | returns and removes item present at the given index |
| [clear()](https://www.programiz.com/python-programming/methods/list/clear) | removes all items from the list |
| [index()](https://www.programiz.com/python-programming/methods/list/index) | returns the index of the first matched item |
| [count()](https://www.programiz.com/python-programming/methods/list/count) | returns the count of the specified item in the list |
| [sort()](https://www.programiz.com/python-programming/methods/list/sort) | sort the list in ascending/descending order |
| [reverse()](https://www.programiz.com/python-programming/methods/list/reverse) | reverses the item of the list |
| [copy()](https://www.programiz.com/python-programming/methods/list/copy) | returns the shallow copy of the list |

**Nested List**

A list within another list is referred to as a **nested list** in Python. We can also say that a list that has other lists as its elements is a nested list. When we want to keep several sets of connected data in a single list, this can be helpful. Here is an illustration of a Python nested list:

MyList = [[22, 14, 16], ["Joe", "Sam", "Abel"], [True, False, True]]

Exercise

1. Write a Python program to count the number of strings from a given list of strings. The string length is 2 or more and the first and last characters are the same.
2. Write a Python program to print the numbers of a specified list after removing even numbers
3. Write a Python program to find the second largest number in a list.
4. Write a Python program to get the frequency of elements in a list.
5. Write a Python program to check whether a list contains a sublist.
6. Write a Python program to convert a list of multiple integers into a single integer.