Experiment 7 :

Aim :- Write a Program to map two lists into a dictionary

Theory:-

write a Python program to map two lists into a dictionary.

A list is an arranged collection of elements. It is used to store collections of data. It can contain a list of various types of data objects with a comma separated and enclosed within a square bracket. It is mutable, which means we can change the order of elements. Individual elements can be replaced even after the list has been created. This is one of the leading reasons why lists are being widely used. Lists play an effective role in small projects as they consume more memory.

Dictionary is an ordered collection of data values, whose values are accessible by key. The dictionary is a mapping of keys and values. The dict() is a constructor which is used to create instances of the class dict. The order of elements in a dictionary is undefined. We can iterate over keys, values, or key-value pairs. Here, we have mentioned different ways to map two lists into a dictionary.

Map two lists into a dictionary using zip() function
The zip() function of Python creates an iterator that aggregates elements from at least two lists. To map two lists together, we can use the Python zip() function. This function allows us to combine two lists together. We can use one list as the keys for the dictionary and the other as the values.

In the given example, we have two lists, one containing a list of students and the other containing their marks. The requirement is to create a single dictionary that stores the name of a student along with their marks. We can use the given solution to accomplish this task:

```
students = ['Smith', 'John', 'Priska', 'Abhi']
marks = [89, 53, 92, 86]

students_dict = dict(zip(students, marks))
print(students_dict)
```

Output :
{'Smith': 89, 'John': 53, 'Priska': 92, 'Abhi': 86}
If the lists vary in size, this method will truncate the longer list. If the keys are not unique, this method will pick the last value for mapping.