|  |
| --- |
|  |
|  |  |
|  | #Q.1. Create a list of the 10 elements of four different types of Data Type like int, string, complex and float. |
|  |  |
|  | mylist = [1,2,3, “sazzz”, “52.8”, “70.7” , 8+4j,3-20j,88,”Job”] |
|  | print(mylist) |
|  |  |
|  | --------------------------------------------------------------------- |
|  | #Q.2 Create a list of size 5 and execute the slicing structure |
|  |  |
|  | l = [2,4,6,8,10] |
|  |  |
|  | print(l[2]) |
|  | print(l[-1]) |
|  | --------------------------------------------------------------------- |
|  |  |
|  | #Q.3 Write a program to get the sum and multiply of all the items in a given list. |
|  |  |
|  | mylist = [1,2,3,4,5] |
|  |  |
|  | Sum, mult= 0 ,1 |
|  |  |
|  | for i in mylist: |
|  | sum = sum + i |
|  | print(sum) |
|  |  |
|  |  |
|  | for i in mylist: |
|  | mul = mul \* i |
|  | #print(mul) |
|  | print(mul) |
|  |  |
|  | --------------------------------------------------------------------- |
|  | #Q.4 Find the largest and smallest number from a given list. |
|  |  |
|  | mylist = [777,532,7544,342,444,123,864,3,45,32,98,54,2,67,1] |
|  |  |
|  | print(max(mylist)) |
|  | print(min(mylist)) |
|  |  |
|  | --------------------------------------------------------------------- |
|  | # Q.5 Create a new list which contains the specified numbers after removing the even numbers from a predefined list. |
|  |  |
|  | mylist = [1,2,3,4,5,6,7,8,9,10,11,1,2,13,14,15] |
|  |  |
|  | newlist = [x for x in mylist if x%2 != 0] |
|  | print(newlist) |
|  |  |
|  | for i in mylist: |
|  | if i%2 == 0: |
|  | mylist.remove(i) |
|  | print(mylist) |
|  |  |
|  | newlist = [] |
|  |  |
|  | for i in mylist: |
|  | if i%2 != 0: |
|  | newlist.append(i) |
|  | print(newlist) |
|  | --------------------------------------------------------------------- |
|  |  |
|  | #Q .6 Create a list of first and last 5 elements where the values are square of numbers between 1 and30 (both included). |
|  |  |
|  | mylist = [] |
|  |  |
|  | for i in range(1,30): |
|  | mylist.append(i\*\*2) |
|  |  |
|  | print(mylist[:5]) |
|  | #print(mylist[0:-5]) |
|  | print(mylist[-5:]) |
|  |  |
|  | --------------------------------------------------------------------- |
|  | ‘’’Q.7 Write a program to replace the last element in a list with another list. |
|  | Sample data: [[1,3,5,7,9,10],[2,4,6,8]] |
|  | Expected output: [1,3,5,7,9,2,4,6,8] |
|  | ‘’’ |
|  | data = [[1,3,5,7,9,10],[2,4,6,8]] |
|  |  |
|  |  |
|  | data[0][5:] = data[1] |
|  |  |
|  | print(data[0]) |
|  |  |
|  |  |
|  | num1 = [1, 3, 5, 7, 9, 10] |
|  | num2 = [2, 4, 6, 8] |
|  |  |
|  | num1[-1:] = num2 |
|  |  |
|  | print(num1) |
|  |  |
|  | --------------------------------------------------------------------- |
|  | #Q.8 Create a new dictionary by concatenating the following two dictionaries: |
|  |  |
|  | a={1:10,2:20} |
|  | b={3:30,4:40} |
|  | # Expected Result: {1:10,2:20,3:30,4:40} |
|  |  |
|  | dict = {} |
|  |  |
|  | for i in (a,b): |
|  | dict.update(i) |
|  |  |
|  | print(dict) |
|  |  |
|  | --------------------------------------------------------------------- |
|  | ‘’’Q.9 Create a dictionary that contains a number (between 1 and n) in the form(x, x\*x). |
|  |  |
|  | Expected Output: {1:1,2:4,3:9,4:16,5:25} |
|  | ‘’’ |
|  | num = int(input("Enter Number: ")) |
|  | dict = {} |
|  |  |
|  | for a in range(1,num+1): |
|  | dict.update({a:a\*a}) |
|  |  |
|  | print(dict) |
|  |  |
|  | --------------------------------------------------------------------- |
|  | ‘’’Q.10 Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple |
|  | which contains every number. Suppose the following input is supplied to the program: |
|  | 34,67,55,33,12,98 |
|  | The output should be: |
|  | [‘34’,’67’,’55’,’33’,’12’,’98’] |
|  | (‘34’,’67’,’55’,’33’,’12’,’98’) |
|  | ‘’’ |
|  | values = input("Enter numbrs seprated by comma: ") |
|  | mylist = values.split(",") |
|  | tuple = tuple(mylist) |
|  |  |
|  | print('List : ',mylist) |
|  | print('Tuple : ',mytuple) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |