**pp 66-71**

**Exercises 1-100 odd**

In Exercises 1 through 48, assume that the list states contain the names of the fifty U.S. states in the order they joined the union, and determine the output displayed by the lines of code.

**states = ["Delaware", "Pennsylvania", "New Jersey", "Georgia", "Connecticut", "Massachusetts", . . ., "Utah", "Oklahoma", "New Mexico", "Arizona", "Alaska", "Hawaii"]**

1. >>>print(states[1], states[-1])

Solution: **Pennsylvania Hawaii**

3.>>> print(states[48], states[49])

Solution: **IndexError: list index out of range**

5. >>>print(states[0], states[-50])

Solution: **IndexError: list index out of range**

7. >>>print(states.index("Alaska"))

Solution: **46**

9. >>>print(states[states.index("Ohio")])

Solution: **ValueError: 'Ohio' is not in list**

11. >>> states[0] = states[0].upper()

>>>

Solution: **DELAWARE**

13. >>>states.append(["Puerto Rico"])

>>>print(states[50])

Solution**: IndexError: list index out of range**

15. >>>states.insert(0, "United States")

>>>print(states[0])

Solution**: United States**

17. >>> print(states[2:5])

Solution: **['New Jersey', 'Georgia', 'Connecticut']**

19. >>>print(states[-5:-2])

Solution: **['Oklahoma', 'New Mexico', 'Arizona']**

21. >>>print(states[:4])

Solution**: ['Delaware', 'Pennsylvania', 'New Jersey', 'Georgia']**

23. >>>print(states[-3:])

Solution: **['Arizona', 'Alaska', 'Hawaii']**

25. >>> print(states[3:3])

Solution**: [ ]**

27. >>> print(states[1:10][2])

Solution: **Georgia**

29. >>> print(states[-2:len(states)])

Solution: **['Alaska', 'Hawaii']**

31. >>> print(states[:][-4])

Solution: **New Mexico**

33. >>>print(len(states[10:20]))

Solution: **10**

35. >>> print(len([]))

Solution: **0**

37. >>>print(len(states[1:-1]))

Solution: **46**

39. >>>states.extend(["Puerto Rico", "Guam"])

>>>print(states[-3:])

Solution: **['Hawaii', 'Puerto Rico', 'Guam']**

41. >>> states += ["Puerto Rico", "Guam"]

>>>print(states[-3:])

Solution: **['Hawaii', 'Puerto Rico', 'Guam']**

43. >>>states[1] = "Commonwealth of Pennsylvania"

>>>print(states[:3])

Solution**: ['Delaware', 'Commonwealth of Pennsylvania', 'New Jersey']**

45. >>>print(states[-4].split())

>>>print(states[2].split())

Solution:

**['New', 'Mexico']**

**['New', 'Jersey']**

47. >>>print((',').join(states[1:4]))

Solution: **Pennsylvania,New Jersey,Georgia**

**In Exercises 49 through 54, assume that list1 contains 100 items. Determine the number of items in each of the slices.**

49. >>>list1[-8:]

Solution: **8**

51. >>>list1[:]

Solution: **100**

53. >>> list1[8:8]

Solution: **Zero items as the list is empty**

**In Exercises 55 through 58, assume that the list nums = [6, 2, 8, 0], and determine the output displayed by the line of code.**

55. >>> print("Largest Number:", max(nums))

Solution: **8**

57. >>> print("Total:", sum(nums))

Solution**: 16**

**In Exercises 59 through 94, determine the output displayed by the lines of code.**

59.

>>>L = ["sentence", "contains", "five", "words."]

>>>L.insert(0, "This")

>>>print(" ".join (L))

**This sentence contains five words.**

>>>del L[3]

>>>L.insert(3, "six")

>>>L.insert(4, "different")

>>>print(" ".join (L))

**This sentence contains six different words**.

61.

>>> name = input("Enter name with two parts: ")

>>>L = name.split() print("{0:s}, {1:s}".format(L[1], L[0]))

**Babbage, Charles**

63.

>>>name = input("Enter name with three parts: ")

**Enter name with three parts: Guido van Rossum**

>>>L = name.split()

>>>print("Middle Name:", L[1])

**Middle Name: van**

65.

>>>tuple1 = ("course", "of", "human", "events", "When", "in", "the")

>>>tuple2 = tuple1[4:] + tuple1[:4]

>>>print((" ".join(tuple2)))

**When in the course of human events**

67.

>>>headEditor = ["editor", "in", "chief"]

>>>print(('-').join(headEditor))

**editor-in-chief**

69.

>>>motto = ["e", "pluribus", "unum"]

>>>print(("\*\*").join(motto))

**e\*\*pluribus\*\*unum**

71.

>>>state = "New York,NY,Empire State,Albany"

>>>stateFacts = state.split(',')

>>>print(stateFacts)

**['New York', 'NY', 'Empire State', 'Albany']**

73.

>>>nations = "France\nEngland\nSpain\n"

>>>countries = nations.split()

>>>print(countries)

**['France', 'England', 'Spain']**

75.

# The three lines of Dev.txt contain mer, gram, pro

>>>infile = open("Dev.txt", 'r')

>>>dev = [line.rstrip() for line in infile]

>>>infile.close()

>>>dev[0], dev[-1] = dev[-1], dev[0]

>>>word = ("").join(dev)

>>>print(word)

**grampromer**

77.

# The three lines of Star.txt contain your, own, star.

>>>infile = open("Star.txt", 'r')

>>>words = [line.rstrip() for line in infile]

>>>infile.close()

>>>words.insert(0, "Follow")

>>>quote = (" ").join(words)

>>>print(quote)

**Follow your own star.**

79.

>>>phoneNumber = "9876543219"

>>>list1 = list(phoneNumber)

>>>list1.insert(3, '-')

>>>list1.insert(7, '-')

>>>phoneNumber = "".join(list1)

>>>print(phoneNumber)

**987-654-3219**

81.

>>>nums = (3, 9, 6)

>>>print(list(nums))

**[3, 9, 6]**

83.

>>>word = "etch"

>>>L = list(word)

>>>L[1] = "a"

>>>print("".join(L))

**each**

85.

>>>list1 = ["soprano", "tenor"]

>>>list2 = ["alto", "bass"]

>>>list1.extend(list2)

>>>print(list1)

**['soprano', 'tenor', 'alto', 'bass']**

87.

>>>list1 = ["gold"]

>>>list2 = ["silver", "bronze"]

>>>print(list1 + list2)

**['gold', 'silver', 'bronze']**

89.

>>>list1 = ["mur"] \* 2

>>>print("".join(list1))

**murmur**

91.

>>>t = ("Dopey", "Sleepy", "Doc", "Grumpy", "Happy", "Sneezy", "Bashful")

>>>print(t[4:20])

**('Happy', 'Sneezy', 'Bashful')**

93.

>>>answer = ["Yes!", "No!", "Yes!", "No!", "Maybe."]

>>>num = answer.index("No!")

>>>print(num)

**1**

I**n Exercises 95 through 99, identify all errors.**

95.

threeRs = ["reading", "riting", "rithmetic"]

print(threeRs[3])

ERROR: IndexError: list index out of range

97.

list1 = [1, "two", "three", 4]

print(" ".join(list1))

ERROR: TypeError: sequence item 0: expected str instance, int found

99. title = ("The", "Call", "of", "the", "Wild")

title[1] = "Calm"

print(" ".join(title))

ERROR: TypeError: 'tuple' object does not support item assignment

101. Analyze a Sentence Write a program that counts the number of words in a sentence input by the user.

Solution:

>>>name=input("Enter a sentence: ")

**Enter a sentence: Know what I mean?**

>>>words=len(name.split())

>>>letters=len(name.replace(' ',''))

>>>avg=letters/words

>>>print("Number of words: ",round(avg))

**Number of words: 4**

102. Analyze a Sentence Write a program that displays the first and last words of a sentence input by the user. See Fig. 2.31. Assume that the only punctuation is a period at the end of the sentence.

Solution:

>>>name=input("Enter a sentence: ")

**Enter a sentence: Reach for the stars.**

>>>entirename=name.split(' ')

>>>fname=entirename[0]

>>>lname=entirename[-1]

>>>print("First word: "+str(fname))

**Reach**

>>>print("Last word: "+str(lname))

**stars**

103. Write a program that requests a two-part name and then displays the name in the form "lastName, firstName".

Solution:

>>>name=input("Enter a two part name: ")

**Enter a two part name: john doe**

>>>entirename=name.split(' ')

>>>fname=entirename[0]

>>>lname=entirename[-1]

>>>print(str(lname)+", "+str(fname))

**Doe, john**