

03_LTD_exercise_solution.py

```
1
2 #####
3 #
4 # [1]. 0, 1, 3, 4, 5, 6, 7, 8 8개의 숫자로 구성된 리스트를 만드시오.
5 #
6 #####
7
8 n = [0, 1, 3, 4, 5, 6, 7, 8]
9 print(n)
10
11 #####
12 #
13 # [2]. 위의 리스트 n의 세 번째 위치에 숫자 2를 삽입하는 파이썬 코드를 작성하시오.
14 #
15 #####
16
17 n.insert(2, 2)
18 print(n)
19
20 #####
21 #
22 # [3]. 위의 리스트 n의 가장 마지막 위치에 숫자 9를 삽입하는 파이썬 코드를 작성하시오.
23 #
24 #####
25
26 n.append(9)
27 print(n)
28
29 #####
30 #
31 # [4]. 위의 리스트 n에서 숫자 5를 삭제하는 파이썬 코드를 작성하시오.
32 #
33 #####
34
35 del n[5]
36
37 # 또는
38 # del n[n.index(5)] # 리스트 n에서 원소 5의 인덱스를 찾아서(n.index(5)) 해당 원소를 삭제
39 # (del)
40 print(n)
41
42 #####
43 #
44 # 아래 코드는 위에서 생성한 리스트 n을 무작위로 섞는 코드입니다. 문제 5번과 6번을 위해 필요하
45 # 니
46 # 그대로 남겨 두세요.
47 #####
48 import random
49
50 random.shuffle(n)
```

```

51 print('n(rearranged):', n)
52
53
54 #####
55 #
56 # [5]. min(), max() 함수를 이용해서 위의 리스트 n에서 최대값과 최소값을 구하시오.
57 #
58 #####
59
60 minN = min(n)
61 maxN = max(n)
62
63 print('minN=', minN)
64 print('maxN=', maxN)
65
66 #####
67 #
68 # [6]. 리스트의 sort(), reverse() 함수를 이용하여 위의 리스트 n을
69 # 내림차순(큰 값이 앞으로, 작은 값이 뒤로)으로 정렬하시오.
70 #
71 #####
72
73 n.sort()      # 오름차순 정렬
74 n.reverse()   # 뒤집기
75 print(n)
76
77
78 #####
79 #
80 # [7]. 아래는 구구단의 2단을 리스트로 구성한 것이다. 이와 같은 방식으로 구구단의 3, 4, 5단을
81 # 리스트로 표현하시오.
82 #
83 # numberMatrix2 = [2, 4, 6, 8, 10, 12, 14, 16, 18]
84 #
85 #####
86 numberMatrix2 = [2, 4, 6, 8, 10, 12, 14, 16, 18]
87 numberMatrix3 = [3, 6, 9, 12, 15, 18, 21, 24, 27]
88 numberMatrix4 = [4, 8, 12, 16, 20, 24, 28, 32, 36]
89 numberMatrix5 = [5, 10, 15, 20, 25, 30, 35, 40, 45]
90
91 #####
92 #
93 # [8]. 위에서 생성한 구구단 2~5단을 하나의 리스트로 표현하여 구구단 리스트를 만드시오.
94 #
95 #####
96
97 numberMatrix = [numberMatrix2, numberMatrix3, numberMatrix4, numberMatrix5]
98
99 # 아래와 같이 코딩하셔도 됩니다.
100 #
101 # numberMatrix = []
102 # numberMatrix.append(numberMatrix2)
103 # numberMatrix.append(numberMatrix3)

```

```

104 # numberMatrix.append(numberMatrix4)
105 # numberMatrix.append(numberMatrix5)
106 #
107 # 많이 사용하는 코딩 패턴들을 위와 같습니다.
108 # 이 외에도 여러 방법들이 더 있을 수 있습니다.
109
110 print(numberMatrix)
111
112 #####
113 #
114 # [9]. 문제 [8]에서 생성한 구구단에서 4단을 화면에 출력하는 파이썬 코드를 작성하시오.
115 #
116 #####
117
118 print(numberMatrix[2])
119
120 # 만약 4단의 3열의 값을 얻으려는 경우 아래와 같이 단 안에서 열 인덱스를 지정하면 됩니다.
121 print(numberMatrix[2][2])
122
123
124 #####
125 #
126 # [10]. 아래 정보를 파이썬 딕셔너리 형태로 표현하고, 생성한 딕셔너리를 화면에 출력하시오.
127 #
128 # '음료': 자몽 주스, 라임 소다
129 # '빵': 모닝롤, 사과 파이
130 # '면': 불닭볶음면, 진라면
131 #
132 #####
133
134 products = {'음료': ['자몽 주스', '라임 소다'], '빵': ['모닝롤', '사과 파이'], '면': ['불닭볶음면', '진라면']}
135
136 # 또는
137 #
138 # products = {}
139 # products["음료"] = ['자몽 주스', '라임 소다']
140 # products["빵"] = ['모닝롤', '사과 파이']
141 # products["면"] = ['불닭볶음면', '진라면']
142
143 print(products)
144
145 #####
146 #
147 # [11]. 문제 [10]에서 생성한 딕셔너리에서 '빵' 카테고리에 해당하는 상품들을 화면에 출력하시오.
148 #
149 #####
150
151 print(products['빵'])
152 # 또는
153 print(products.get('빵'))
154 # 딕셔너리에서 제공하는 get() 함수를 사용하면 딕셔너리 안에 키가 존재하지 않는 경우에도
155 # 오류가 발생하지 않고 대신 None 값이 반환됩니다.
156

```

```
157 #####
158 #
159 # [12]. 문제 [10]에서 생성한 딕셔너리의 '빵' 카테고리에 아래 상품을 추가하고 '빵' 카테고리의
    상품들을 다시 출력하시오.
160 #
161 # 슈크림 크로와상
162 #
163 #####
164
165 products['빵'].append('슈크림')
166 products['빵'].append('크로와상')
167
168 # 또는
169 # products['빵'] = products['빵'] + ['슈크림', '크로와상']
170
171 print(products['빵'])
172
173
174
175
```