# 14차 연습문항

## **파이썬 추가문항: 자료구조 시퀸스**

|  |
| --- |
| 수강생 공지 사항   * 제출은 프로그래머스를 통해 해주시기 바랍니다([링크](https://campus.programmers.co.kr/app/courses/24550/curriculum)). * 파일명은 아래와 같은 형식으로 제출해주세요   + 교육생번호\_이름\_교과목\_문항\_N차시\_강의명.pdf   ex) DR-11111\_홍길동\_파이썬\_연습문항\_1차시\_환경및기본.pdf  ex) DR-11111\_홍길동\_파이썬\_추가문항\_1차시\_환경및기본.pdf   * 답은 "write your answer"에 적어주세요. 다만 코딩 문제의 경우 output까지 답에 포함시켜 주세요.   ex)     * 답은 캡쳐를 하셔도 되고 텍스트로 넣으셔도 됩니다. * 마감 기한은 문제가 나간 주 **일요일 23:59까지**입니다. |

### 아래 프로그램을 리스트 컴프리헨션을 사용한 코드로 변환하시오.

|  |
| --- |
| Example Code |
| even = []  for i in range(2, 11, 2):        even.append(i)  print(even)  # Write your code here: 위 코드를 List comprehension으로 변환하시오. |

**답**

|  |
| --- |
| even = [i for I in range(2, 11,2)]  print(even) |

### 아래와 같이 if 조건문이 있는 반복문으로 리스트를 생성하는 프로그램을 리스트 컴프리헨션을 적용한 프로그램으로 작성하시오.

|  |
| --- |
| Example Code |
| s = []  for i in range(10):        if i % 2 == 1:              s.append(i\*\*2)  print(s)  # Write your code here: 위 코드를 List comprehension으로 변환하시오. |

**답**

|  |
| --- |
| s = [i\*\*2 for i in range(10) if i % 2 == 1]  print(s) |

### 아래의 프로그램을 실행한 후 결과를 출력해 보시오.

|  |
| --- |
| Example Code |
| mylist = [1, 5, 1, 7, 1]  mylist[mylist.count(1)] = 70  mylist[len(mylist) - 1] = 80  mylist.insert(1, 50)  print(mylist) |

**답**

|  |
| --- |
| [1, 50, 5, 1, 70, 80] |

### 1에서 99까지의 난수 10개로 리스트를 만든 후 리스트와 정렬된 리스트 그리고 내림차순으로 정렬된 역순 리스트를 출력하는 프로그램을 작성하시오..

**답**

|  |
| --- |
| random\_list = [random.randint(1, 99) for \_ in range(10)]  # 원본 리스트 출력  print("원본 리스트:", random\_list)    # 정렬된 리스트 출력  sorted\_list = sorted(random\_list)  print("정렬된 리스트:", sorted\_list)    # 내림차순으로 정렬된 역순 리스트 출력  reverse\_sorted\_list = sorted(random\_list, reverse=True)  print("내림차순으로 정렬된 역순 리스트:", reverse\_sorted\_list) |

### 영한사전과 같이 한글과 영어에 대응되는 튜플 korean과 english를 만든 후, 표준입력으로 한글을 입력받아 영어를 출력하는 프로그램을 작성하시오.

korean = ('정렬‘, ’문자’, ‘내포’ , ‘사전’)

english - ('sorting', 'text', 'comprehension', 'dictionary')

- 단어가 사전에 없으면 적절한 메시지를 출력할 것

**답**

|  |
| --- |
| def translate\_word(word):        korean = ('정렬', '문자', '내포', '사전')        english = ('sorting', 'text', 'comprehension', 'dictionary')        try:              index = korean.index(word)              return english[index]        except ValueError:              return "해당 단어를 찾을 수 없습니다."    def main():        # 사용자로부터 한글 단어 입력 받기        korean\_word = input("한글 단어를 입력하세요: ")        # 단어 번역        translated\_word = translate\_word(korean\_word)        # 결과 출력        print(f"{korean\_word} -> {translated\_word}")    if \_\_name\_\_ == "\_\_main\_\_":       # 스크립트 파일이 직접 실행될 때만 main() 함수를      main()                                          # 호출하도록 하는 조건문 |

### 다음 중첩된 리스트 data에서 각 행의 합과 열의 합을 리스트 rsum과 csum에 저장해 출력하는 프로그램을 작성하시오.

**답**

|  |
| --- |
| data = [        [1, 2, 3],        [4, 5, 6],        [7, 8, 9]  ]  # 행의 합을 저장할 리스트  rsum = []  # 열의 합을 저장할 리스트, 처음에 0으로 초기화  csum = [0] \* len(data[0])  # 각 행의 합 계산  for row in data:        row\_sum = sum(row)        rsum.append(row\_sum)        # 각 열의 합 계산        for i in range(len(row)):              csum[i] += row[i]    # 결과 출력  print("각 행의 합:", rsum)  print("각 열의 합:", csum) |

### 아래의 리스트에서 리스트 컴프리헨션을 활용해 행과 열이 바뀐 형태의 리스트를 새로 만들고, 이 변환된 리스트를 다음과 같이 출력하는 프로그램을 작성하시오.

- 다음 리스트 컴프리헨션을 활용하세요.

transpose = [[row[i] for row in m] for I in range(len(m[0]))]

**답**

|  |
| --- |
| # 주어진 리스트  m = [[1, 2], [3, 4], [5, 6], [7, 8]]    # 리스트 컴프리헨션을 사용하여 행과 열을 바꾼 새로운 리스트 생성  transpose = [[row[i] for row in m] for i in range(len(m[0]))]    # 결과 출력  for row in transpose:        print(row) |

### 1에서 20까지의 난수 5개를 두 번 얻어 각각 집합인 변수 A, B에 저장한다. 집합 A와 B의 합집합과 교집합, 차집합, 여집합을 구해 출력하는 프로그램을 작성하시오.

- 모듈 random의 sample() 함수를 사용한 다음 문장을 집합 변수 A에 저장

**답**

|  |
| --- |
| import random  # 1에서 20까지의 난수 5개를 두 번 얻어 각각 집합 A와 B에 저장  A = set(random.sample(range(1, 21), 5))  B = set(random.sample(range(1, 21), 5))  # 합집합  union = A | B  # 교집합  intersection = A & B  # 차집합 (A - B)  difference\_A\_B = A - B  # 차집합 (B - A)  difference\_B\_A = B - A  # 여집합 (전체 집합에서 B를 뺀 나머지, 여기서는 1에서 20까지의 범위)  universal\_set = set(range(1, 21))  complement\_B = universal\_set - B  # 결과 출력  print("집합 A:", A)  print("집합 B:", B)  print("합집합 (A | B):", union)  print("교집합 (A & B):", intersection)  print("차집합 (A - B):", difference\_A\_B)  print("차집합 (B - A):", difference\_B\_A)  print("여집합 (전체 집합 - B):", complement\_B) |