|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 8차시 | 1 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제1) 리스트에 대한 설명 중 맞지 않은 것을 고르시오.  (1) 리스트는 순서가 있는 변경 가능한 자료형이다.  **(2) 리스트는 중복된 값을 가질 수 없다.**  (3) 리스트는 다양한 자료형을 요소로 가질 수 있다.  (4) 리스트는 슬라이싱을 통해 부분 리스트를 추출할 수 있다.    문제2) 다음 중 리스트 요소를 구성하는 방법으로 맞지 않는 것을 모두 고르시오.  (1) [1, 2, 3, [1, 2]]  (2) [ 1, 'h', "ab", 2, 5 ]  **(3) [1, , 3]**  (4) [1, '', 3]  **- (4)의 ''는 빈 문자열을 나타낸다. 요소로 사용하는데 문제없다**  문제3) 다음과 같이 리스트를 생성할 수 있는지 확인하고, 출력 내용을 작성하시오.  lst2 = list()  print(lst2)  **정답3) 생성 가능하다. 빈 리스트가 생성되고 출력된다.**  **출력내용**  **[]**  문제4) 다음과 같이 문자열 리스트를 반복문으로 출력해보세요. 문자가 하나씩 출력이 되는지? 코드를 수정해서 한 글자씩 출력해보세요.  for ch in ["hello"]:  print(ch)  **정답4)**  **for ch in ["hello"]:**  **print(ch)**  **hello**  **["hello"]는 요소가 "hello"인 리스트이다. 따라서 ch에는 문자열 "hello"가 저장되고, 문자열 전체가 출력된다.**  **- 한 글자씩 출력하는 두 가지 방법**  **(1) 리스트의 요소 문자열을 인덱스를 이용해서 추출해서 for 반복문에 적용**  **lst = ["hello"]**  **for ch in lst[0]:**  **print(ch)**  **(2) 중복 반복문을 사용. 단 두번째 방법은 lst의 요소들이 모두 문자열 또는 리스트로 구성되어 있어야 한다.**  **lst = ["hello"]**  **for element in lst:**  **for ch in element:**  **print(ch)** | | | | | | |

|  |
| --- |
| 문제5) 다음 중 ValueError가 발생하는 이유로 올바른 것을 고르세요.  (1) 리스트의 존재하지 않는 인덱스에 접근하려고 할 때 **(2) 문자열을 정수로 변환하려고 할 때** (3) 튜플의 요소를 변경하려고 할 때 (4) 정수를 0으로 나누려고 할 때    문제6) 다음 결과를 적으시오.  (1) "He is not my type, \tthanks".split(" \t")  **(1) ['He is not my type,', 'thanks']**  (2) "He is not my type, \tthanks".split(" ", 2)  **(2) ['He', 'is', 'not my type, \tthanks']**  (3) "He is not my type, \tthanks".split("t")  **(3) ['He is no', ' my ', 'ype, \t', 'hanks']**  문제7) 튜플을 잘못 사용한 예를 고르시오.  (1) my = ()  (2) my = tuple()  (3) my = (1, )  **(4) my = (1) : my는 요소 1을 가지는 튜플이 아니라 정수값 1이 저장된 정수형 변수가 된다. 튜플로 지정하기 위해선 (1,)와 같이 작성해야함.**  (5) my = (1, 2, [3, 4])    문제8) 다음을 실행하면 my에 어떤 변화가 생기는지 작성하시오. 오류가 발생하면 이유를 설명하시오.  my = (1, 2)  my.append([3, 4])  **정답8) 오류 발생. 튜플에는 append() 함수가 포함되어 있지 않다.**  문제9) my1, my2를 다음과 같이 정의하자.  my1=(1,2,[3,4])  my2=(1,2,(3,4))  다음 연산이 가능한지 OX로 답하시오  (1) my1[2][0] = 33  **(1) O 리스트 [3, 4]의 첫 번째 요소인 3이 33으로 바뀜**  (2) my2[2][0] = 33  **(2) X 튜플의 세 번째 요소는 튜플이고 변경 불가능함**  문제10) 아래의 리스트를 튜플로 변환하고, 다시 리스트로 변환하는 코드를 작성하시오.  my\_list = [1, 2, 3, 4]  **정답10)**  **my\_list = [1, 2, 3, 4]**  **my\_tuple = tuple(my\_list) # 리스트를 튜플로 변환**  **print(my\_tuple)**  **my\_tuple = (1, 2, 3, 4)**  **my\_list = list(my\_tuple) # 튜플을 리스트로 변환**  **print(my\_list)** |

|  |
| --- |
| 문제11) 정수 n을 인자로 받아, n의 모든 약수를 리스트로 만들어 반환하는 함수를 구현하시오. 이 함수를 이용하여 2~20까지의 정수에 대해 약수의 개수를 다음과 같이 출력하는 코드를 작성하시오.    2의 약수 개수: 2  3의 약수 개수: 2  ...  19의 약수 개수: 2  20의 약수 개수: 6    **정답11)**  **def createDivisorsList(n):**  **lst = []**  **for i in range(1, n + 1):**  **if n % i == 0:**  **lst.append(i)**  **return lst**  **for i in range(2, 21):**  **lst = createDivisorsList(i)**  **print(f"{i}의 약수 개수: {len(lst)}"**  문제12) 주어진 리스트에서 중복된 요소를 제거하시오. 단 순서는 그대로 유지할 것.  myList = [1, 2, 2, 3, 4, 4, 5]  **정답12)**  **myList = [1, 2, 2, 3, 4, 4, 5]**  **def remove\_duplicates(lst):**  **unique\_lst = []**  **for item in lst:**  **if item not in unique\_lst:**  **unique\_lst.append(item)**  **return unique\_lst**  **# 테스트**  **print(remove\_duplicates(myList))** |

|  |
| --- |
| 문제13) 정수 n1과 n2를 인자로 받아 n1부터 n2까지 각 정수의 약수를 화면에 출력하는 함수를 구현하시오. 그리고 이 함수를 이용하여 10부터 16까지의 약수를 출력해 보시오.  **정답13)**  **def printDivisors(n1, n2):**  **for n in range(n1, n2 + 1):**  **print(f"n = {n}")**  **for i in range(1, n + 1): # 약수 출력**  **if n % i == 0:**  **print(i, end = ' ')**  **print(printDivisors(10, 16))**  **또는  def printDivisors(n1, n2):**  **"""n1부터 n2까지 각 정수의 약수를 화면에 출력하는 함수"""**  **def get\_divisors(n):**  **"""정수 n의 모든 약수를 리스트로 반환하는 함수"""**  **divisors = [i for i in range(1, n + 1) if n % i == 0]**  **return divisors**  **for i in range(n1, n2 + 1):**  **divisors = get\_divisors(i)**  **print(f"{i}의 약수: {divisors}")**  **# 10부터 16까지의 각 정수의 약수를 출력**  **printDivisors(10, 16)**  문제14) 오름차순으로 정렬된 두 개의 정수 리스트를 입력으로 받아, 이 두 리스트를 하나의 오름차순으로 정렬된 리스트로 합친 후 반환하는 함수를 구현하시오. 단 함수에 인자로 전달되는 두 개 리스트는 길이가 다를 수 있음.  이 함수를 사용하여 [1, 5, 8, 10, 14]와 [2, 4, 5, 9]를 한 개의 정렬된 리스트로 만들고, 이를 화면에 출력하는 프로그램을 작성하시오. 함수가 반환하는 리스트는 [1, 2, 4, 5, 5, 8, 9, 10, 14]이다.  **정답14)**  **def mergeLists(lst1, lst2):**  **# 두 개 리스트의 길이를 구함**  **len1 = len(lst1)**  **len2 = len(lst2)**  **# 두 리스트의 현재 인덱스를 저장할 변수를 선언**  **idx1 = 0**  **idx2 = 0**  **# 두 리스트를 합친 데이터를 담을 리스트 생성**  **newLst = []**  **# 두 개 리스트를 비교하면서 새로운 리스트로 정렬**  **while idx1 < len1 and idx2 < len2:**  **# 작은 쪽을 리스트에 추가**  **if lst1[idx1] <= lst2[idx2]:**  **newLst.append(lst1[idx1])**  **idx1 += 1**  **else:**  **newLst.append(lst2[idx2])**  **idx2 += 1**  **# 여기까지 왔다는 것은 두 리스트 중 한 개의 모든 요소가 새로운 리스트에 추가된 것임**  **# 따라서 남은 리스트의 요소들을 추가하면 됨**  **for i in range(idx1, len1):**  **newLst.append(lst1[i])**  **for i in range(idx2, len2):**  **newLst.append(lst2[i])**  **return newLst**    **print(mergeLists([1,5,8,10,14], [2,4,5,9]))** |

|  |
| --- |
| 문제15) 정수 n을 함수의 인자로 받아, 사용자로부터 n개의 정수를 입력받아 리스트로 구성한 후 반환하는 함수를 작성하시오. 이 함수를 사용하여 n개의 정수로 구성된 리스트를 생성하고 출력하는 프로그램을 작성하시오.  <요구사항>  - 사용자로부터 입력받을 정수의 개수는 함수에 인자로 전달  - 사용자로부터 입력받는 값은 양의 정수로 가정  **정답15)**  **def createListFromInput(n):**  **lst = []**  **for i in range(n):**  **num = int(input(f"정수를 입력하세요 ({i+1}/{n}): "))**  **lst.append(num)**  **return lst**    **lst1 = createListFromInput(int(input("리스트의 인자의 개수를 입력하세요: ")))**  **print(lst1) # 리스트를 확인차 출력**    **for n in lst1:**  **print(n) # 요소 출력**  문제16) 정수 n을 함수의 인자로 받아, 그 정수의 모든 약수를 리스트로 구성하여 반환하는 함수(createDivisorsList)를 구현하시오. 이 함수를 사용하여 사용자로부터 1부터 1000까지의 정수 중 하나를 입력받고, 그 정수의 모든 약수의 합을 계산하여 출력하는 프로그램을 작성하시오.  - 정수 n의 약수를 찾는 방법은 1~n까지의 정수를 n으로 나눠서 나머지가 0인지 확인  <요구사항>  - 입력 받는 값은 1~1000 정수로 가정  - createDivisorsList함수를 호출하는 코드에서 리스트의 내용을 화면에 출력함.  **정답16)**  **def createDivisorsList(n):**  **lst = []**  **for i in range(1, n + 1):**  **if n % i == 0:**  **lst.append(i)**  **return lst**    **# 사용자로부터 정수 입력 받기**  **n = int(input("1000이하의 정수 한 개를 입력하세요: "))**  **print(createDivisorsList(n))** |

|  |
| --- |
| 문제17) 정수 n을 함수의 인자로 받아, 사용자로부터 n개의 정수를 입력받아 이를 요소로 하는 튜플을 구성하여 반환하는 함수를 작성하시오. 이 함수를 사용하여 n개의 정수로 구성된 튜플을 생성하고 출력하는 프로그램을 작성하시오.  <요구사항>  - 사용자가 입력한 정수의 개수는 함수에 인자로 전달  - 사용자로부터 입력 받는 값은 양의 정수만으로 가정  **정답17)**  **def createTupleFromInput(n):**  **t = ()**  **for i in range(n):**  **num = int(input(f"정수를 입력하세요 ({i+1}/{n}): "))**  **t += (num,)**  **return t**    **n = int(input("튜플의 인자의 개수를 입력하세요"))**  **t1 = createTupleFromInput(n)**  **print(t1) # 튜플의 내용을 확인하기 위해 출력**    **for e in t1:**  **print(e) # 각 요소 출력**  문제18) 도형 정보를 담고 있는 튜플의 요소들을 이용해서 도형의 면적을 계산해서 출력하는 프로그램 작성.  - 도형 정보를 담고 있는 튜플의 예시 (“사각형”, 30, 20, “원”, 10): 사격형의 경우 직사각형이며, 두변의 길이 정보가 뒤이어 나온다. 원의 경우 반지름이 뒤이어 나온다.  <요구사항>  - 튜플에 있는 도형의 개수는 정해져 있지 않음  - 원주율은 math.pi 사용 (import math 필요)  - calcAndPrintArea() 함수는 튜플을 인자로 받고 면적을 계산해서 화면에 출력  - 튜플 예시: ("사각형", 30, 20, "원", 10, "사각형", 20, 40, "사각형", 10, 10,"원", 20) 사용  - 출력 예시: 도형\_종류, 면적계산시 필요정보, 넓이  **정답18)**  **import math**  **def calcAndPrintArea(t):**  **idx = 0**  **while idx < len(t):**  **if t[idx] == "사각형":**  **area = t[idx + 1] \* t[idx + 2]**  **print(f"{t[idx]}, {t[idx + 1]}, {t[idx + 2]}, {area}")**  **idx += 3**  **elif t[idx] == "원":**  **area = t[idx + 1] \*\* 2 \* math.pi**  **print(f"{t[idx]}, {t[idx + 1]}, {area}")**  **idx += 2**  **t = ("사각형", 30, 20, "원", 10, "사각형", 20, 40, "사각형", 10, 10, "원", 20)**  **calcAndPrintArea(t)** |

|  |
| --- |
| 문제19) 다음 중첩된 리스트 data에서 각 행의 합과 열의 합을 리스트 rsum과 sum에 저장해 출력하는 프로그램을 작성하시오.  data = [ [1, 2, 3],  [4, 5, 6],  [7, 8, 9]]    **정답19)**  **data = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]**    **# 행의 합 계산하여 rsum에 저장**  **rsum = []**  **for row in data:**  **row\_sum = sum(row)**  **rsum.append(row\_sum)**    **# 열의 합 계산하여 csum에 저장**  **csum = []**  **num\_cols = len(data[0]) # 데이터의 첫 번째 행의 열의 수를 기준으로 함**  **for col\_idx in range(num\_cols):**  **col\_sum = sum(row[col\_idx] for row in data)**  **csum.append(col\_sum)**    **# 결과 출력**  **print("각 행의 합 (rsum):", rsum)**  **print("각 열의 합 (csum):", csum)**  문제20) 문제19)에 주어진 리스트에서 두 번째로 큰 값을 찾는 함수를 작성하세요. 단, 모든 요소 는 고유한 값이라고 가정합니다.  **정답20)**  **def find\_second\_largest(lst):**  **if len(lst) < 2:**  **return None**  **largest = second\_largest = float('-inf') #부동소수점 형식의 음의 무한대를**  **# 나타내는 특수한 값**  **for num in lst:**  **if num > largest:**  **second\_largest = largest**  **largest = num**  **elif num > second\_largest:**  **second\_largest = num**  **return second\_largest**    **# 테스트**  **print(find\_second\_largest([5, 2, 9, 1, 5])) # 출력: 5**  **print(find\_second\_largest([3, 1, 4, 1, 5, 9, 2])) # 출력: 5** |

|  |
| --- |
| 문제21) 아래의 중첩된 리스트를 for문으로 행과 열을 맞춰 출력한 후, 다시 행과 열을 바꾼 형태로 출력하세요.  lst = [[1,2],[3,4],[5,6],[7,8]]  결과: 1 3 5 7  2 4 6 8  **정답21)**  **lst = [[1, 2], [3, 4], [5, 6], [7, 8]]**  **# 행과 열을 맞춰 원래 리스트 출력**  **print("Original List:")**  **for row in lst:**  **for item in row:**  **print(item, end=' ')**  **print() # 행이 끝나면 줄바꿈**  **# 전치 행렬 생성**  **transposed\_lst = []**  **for i in range(len(lst[0])): # 열의 개수만큼 순회**  **new\_row = []**  **for j in range(len(lst)): # 행의 개수만큼 순회**  **new\_row.append(lst[j][i]) # 각 원소를 새로운 행에 추가**  **transposed\_lst.append(new\_row) # 새로운 행을 전치 리스트에 추가**  **# 행과 열을 바꾼 리스트 출력**  **print("\nTransposed List:")**  **for row in transposed\_lst:**  **for item in row:**  **print(item, end=' ')**  **print() # 행이 끝나면 줄바꿈**  문제22) 1에서 99까지의 난수 10개로 리스트를 만든 후, 다시 이 리스트를 튜플로 변환 하고, 다시 정렬된 리스트로 만들어서 전체의 합, 최대값, 최소값, 평균을 구하는 코드를 작성하시오.    **정답22)**  **import random**  **# 1에서 99까지의 난수 10개로 리스트 생성**  **random\_list = [random.randint(1, 99) for \_ in range(10)] #리스트 컴프리헨션**  **print("Random List:", random\_list)**  **# 리스트를 튜플로 변환**  **random\_tuple = tuple(random\_list)**  **print("Tuple:", random\_tuple)**  **# 튜플을 다시 리스트로 변환하고 정렬**  **sorted\_list = sorted(random\_tuple)**  **print("Sorted List:", sorted\_list)**  **# 전체의 합, 최대값, 최소값, 평균 구하기**  **total\_sum = sum(sorted\_list)**  **max\_value = max(sorted\_list)**  **min\_value = min(sorted\_list)**  **average = total\_sum / len(sorted\_list)**  **print("Sum:", total\_sum)**  **print("Max Value:", max\_value)**  **print("Min Value:", min\_value)**  **print("Average:", average)** |

|  |
| --- |
| 문제23) 사용자로부터 삼각형의 꼭지점 좌표 를 입력 받고 리스트 를 형태로 구성해서 반환하는 함수를 구현한다. 함수에서 반환되는 삼각형 좌표 리스트를 이용해서 삼각형의 면적을 구하는 함수를 구현한다. 두 개 함수를 이용해서 사용자로부터 입력받은 삼각형 좌표를 이용해서 면적을 계산하고 화면에 출력하는 프로그램을 작성한다. 삼각형 좌표를 이용해서 면적을 구하는 것은 헤론의 공식을 이용한다.    헤론의 공식: 삼각형의 세 변의 길이를 각각 a, b, c라고 가정할 때 면적을 구하는 공식    면적 =  **정답23)**  **import math**  **# p1과 p2는 같은 크기라고 가정**  **def calcDistance(p1, p2):**  **sum = 0**  **for i in range(len(p1)):**  **sum += ((p2[i] - p1[i]) \* (p2[i] - p1[i]))**  **return math.sqrt(sum)**  **def createTriangleCoords():**  **x1 = int(input("좌표 1의 x값 입력: "))**  **y1 = int(input("좌표 1의 y값 입력: "))**  **x2 = int(input("좌표 2의 x값 입력: "))**  **y2 = int(input("좌표 2의 y값 입력: "))**  **x3 = int(input("좌표 3의 x값 입력: "))**  **y3 = int(input("좌표 3의 y값 입력: "))**  **return [[x1, y1], [x2, y2], [x3, y3]]**  **def calcTriangleArea(lst):**  **a = calcDistance(lst[0], lst[1])**  **b = calcDistance(lst[1], lst[2])**  **c = calcDistance(lst[2], lst[0])**  **s = (a + b + c) / 2**  **area = math.sqrt(s \* (s - a) \* (s - b) \* (s - c))**  **return area**  **lst = createTriangleCoords()**  **area = calcTriangleArea(lst)**  **print(area)** |