|  |
| --- |
| **py\_sci\_prog\_06\_set\_ex** |

|  |
| --- |
| 1. 사용자로부터 문자를 여러 개 입력 다음과 같이 수행하는 프로그램을 작성하세요.  >>>  문자 여러 개 입력 : qwerqwerqwer  문자 여러 개 입력 : aseraseraseraser  qwerqwerqwer 종류= {'q', 'r', 'w', 'e'}  aseraseraseraser 종류= {'r', 'e', 's', 'a'}  두 종류에 동시에 들어 있는 문자= {'r', 'e'}  >>>  **[소스]**  # 1  firstStr = input("문자 여러개 입력 : ")  secondStr = input("문자 여러개 입력 : ")  s1 = set(firstStr)  s2 = set(secondStr)  print(firstStr, "종류 = ", s1)  print(secondStr, "종류 = ", s2)  if(s1.isdisjoint(s2)):  print("두 종류에 동시에 들어 있는 문자가 없습니다.")  else:  print("두 종류에 동시에 들어 있는 문자 = ", s1.intersection(s2)) |

|  |
| --- |
| 2. 주사위 두 개를 던졌을 때 얻을 수 있는 주사위 눈에 대한 모든 경우의 수를 result\_case 이름의 set에 저장한 후 result\_case 저장된 내용을 출력하는 프로그램을 작성하세요.  >>>  [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6)]  >>>  **[소스]**  # 2  result\_case = sorted(set((i, j) for i in range(1, 7) for j in range(1, 7)))  print(result\_case) |

|  |
| --- |
| 3. 주사위 두 개를 던졌을 때 얻을 수 있는 주사위 눈에 대한 모든 경우의 수에 대해, 주사위 눈의 합의 모든 종류와 중복을 제외한 합의 종류를 출력하는 프로그램을 작성하세요.  >>>  sum\_list [4, 12, 11, 3, 8, 7, 6, 7, 3, 6, 10, 8, 9, 6, 9, 4, 7, 5, 2, 10, 5, 8, 9, 9, 5, 5, 6, 11, 8, 8, 10, 7, 4, 7, 7, 6]  sum\_set {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}  >>>  **[소스]**  # 3  sum\_list = [(i+j) for i in range(1, 7) for j in range(1, 7)]  sum\_set = set(i for i in sum\_list)    print("sum\_list", sum\_list)  print("sum\_set", sum\_set) |

|  |
| --- |
| 4. 다음 출력 값은 무엇입니까?  4-1) list(zip([1,2,1], "ABA")) >> [(1, 'A'), (2, 'B'), (1, 'A')]  4-2) tuple(zip([1,2,1], "ABA")) >> ((1, 'A'), (2, 'B'), (1, 'A'))  4-3) set(zip([1,2,1], "ABA")) >> {(2, 'B'), (1, 'A')} |

|  |
| --- |
| 5. 리스트 a와 튜플 b를 zip() 함수를 이용하여 통합된 하나의 set를 생성하세요.  a = [10, 20, 30]  b = {'twenty', 'ten', 'thirty'}  >>>  a = [10, 20, 30]  b = {'twenty', 'ten', 'thirty'}  zip\_val = {(10, 'twenty'), (20, 'ten'), (30, 'thirty')}  >>>  **[소스]**  # 5  a = [10, 20, 30]  b = {'twenty', 'ten', 'thirty'}  zip\_val = set(zip(a, b))  print(sorted(zip\_val)) |

|  |
| --- |
| 6. 컴프리헨션 형식을 사용하여 1부터 사용자에게 입력 받은 수까지의 5의 배수를 출력하세요.  >>>  어디까지 : 50  [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]  >>>  **[소스]**  # 6  print(sorted(val for val in range(1, int(input("어디까지 : "))+1) if val%5 == 0)) |

|  |
| --- |
| 7. 다음 프로그램의 수행 결과를 눈으로 예측하세요. 실행 결과가 예측과 다르다면 이유를 설명하고 수정하세요.  stra = ['a', 'p', 'p', 'l', 'e']  strb=stra  seta = set(stra)  setb = set(strb)  strb.append('s')  print('stra=', stra)  print('strb=', strb)  **[소스]**  # 7  stra = ['a', 'p', 'p', 'l', 'e']  # strb = stra 를 하면 두개의 리스트가 같은 값을 가르키기 떄문에  # 다른 리스트의 값을 변경해도 둘다 변하기 때문이다  # 그러므로 .copy() 를 사용해서 shallow copy 만 이끌어낸다  # 그렇게 되면 다른 리스트 안에 값을 변경해도 다른 리스트가 같이 변하지 않는다.  strb = stra.copy()  seta = set(stra)  setb = set(strb)  strb.append('s')  print('stra=', stra)  print('strb=', strb) |

|  |
| --- |
| 8. 다음 입력에 대한 괄호의 짝이 올바른지 눈으로 확인하세요.  a) {2 \* { 3 + ( 4 - 5 ) \* 6 + { 7 + ( 8 - 9 ) // 10 } \* 2 } +3 } 올바르다.  b) { 2 \* { 3+ ( 4 – 5 ) \*6 + { 7 + ( 8 - 9 ) //10 } **\*2 ) + 3** + ( 4 ) -5 } 올바르지 않다.  --> \*2 ) + 3 부분이 \*2 } + 3 로 바뀌면 된다 |

|  |
| --- |
| 9. 중괄호와 소괄호 만으로 이루어진 입력 스트링에서 괄호의 짝을 맞추기 위한 프로그램을 작성합니다. 입력은 다음과 같은 astr 처럼 문자열로 주어집니다. 아래와 같은 결과를 얻는 프로그램을 작성하셨다면, 위 9번 문제의 a)와 b)의 예상 결과가 옳은지 프로그램을 통해 확인해 봅니다. (배달의 민족 입사 시험에 나왔던 문제로, 이 문제도 시간복잡도가 최저인 경우 고득점을 받습니다.)  astr = '{2 \* { 3 + ( 4 - 5 ) \* 6 + { 7 + ( 8- 9 ) // 10 } \* 2 } + 3 }'  >>>  astr= {2 \* { 3 + ( 4 - 5 ) \* 6 + { 7 + ( 8- 9 ) // 10 } \* 2 } + 3 }  astr is OK  >>>  >>>  astr= {2 \* { 3 + ( 4 - 5 ) \* 6 + { 7 + ( 8- 9 ) // 10 } \* 2 } +3  astr is NOK  >>>  **[소스]**  # 9  parentheses = ['(', ')', '{', '}', '[', ']']  cnt = dict.fromkeys(parentheses, 0)  myStr = input('str = ')  str\_list = [n for n in myStr if n in cnt.keys()]  for i in str\_list:  if i in cnt.keys():  cnt[i] += 1  cnd = []  for i in range(0, 6, 2):  cnd.append(True if cnt.get(parentheses[i]) == cnt.get(parentheses[i+1]) else False)  if all(cnd):  print("myStr is OK")  else :  print("myStr is NOK") |