|  |
| --- |
| **py\_sci\_prog\_04\_list\_ex** |

20.04/6 p\_04\_김민석\_20195124

|  |
| --- |
| 1. 제시된 조건에 알맞은 프로그램을 작성하세요. 리스트 data = [ 21, 7, 43, 65, 2, 8, 72, 52, 9 ]가 존재 할 때, 리스트의 내용을 거꾸로 출력하세요.  >>>  [9, 52, 72, 8, 2, 65, 43, 7, 21]  >>>  **[소스]**  **# 1-1 :: 범위 이용**  data = [ 21, 7, 43, 65, 2, 8, 72, 52, 9 ]  print(data[::-1])  **# 1-2 reversed 이용**  data = [ 21, 7, 43, 65, 2, 8, 72, 52, 9 ]  print(list(reversed(data))) |

|  |
| --- |
| 2. 제시된 조건에 알맞은 프로그램을 작성하세요. 리스트 data = [ 21, 7, 43, 65, 2, 8, 72, 52, 9 ]가 존재 할 때, 정수를 입력 받아 리스트 내의 값을 검색해 봅니다. 값을 찾은 경우 “위치 : n”으로 출력하고 반복을 계속 합니다. 찾지 못한 경우 “찾지 못함”이라고 출력하고 종료합니다.  >>>  찾을 값? 21  -> 위치: 0  찾을 값? 65  -> 위치: 3  찾을 값? 77  -> 찾지 못함  >>>  **[소스]**  # 2  data = [ 21, 7, 43, 65, 2, 8, 72, 52, 9 ]  while(True):  num = int(input("찾을 값? "))  loc = -1    for n, v in enumerate(data):  print(n, v)  if v == num:  loc = n    if loc != -1:  print(" -> 위치 : ", loc)  else:  print(" -> 찾지 못함 ")  break |

|  |
| --- |
| 3. 제시된 조건에 알맞은 프로그램을 작성하세요. 리스트 data = [ [21, 7, 43, 65], [2, 8, 72, 52] ]가 존재 할 때, 정수를 입력 받아 리스트 내의 값을 검색합니다. 입력 값이 숫자가 아닐 경우에는 다시 입력 받고, 값을 찾은 경우 “위치 : n, m”으로 출력하고 반복을 계속 합니다. 찾지 못한 경우 “찾지 못함”이라고 출력하고 종료합니다. (Hint: 숫자 판별 isdigit())  >>>  찾을 값 : 7  -> 위치: 0 , 1  찾을 값 : a  -> 숫자 아님  찾을 값 : 8  -> 위치: 1 , 1  찾을 값 : 6  -> 찾지 못함  >>>  **[소스]**  # 3  data = [[21, 7, 43, 65], [2, 8, 72, 52]]  while(True):  cnd = True  i = input("찾을 값 : ")  if not i.isdigit():  print(" -> 숫자 아님 ")  continue    for n, v in enumerate(data):  for m, k in enumerate(v):  if int(i) == k:  cnd = False  print(" -> 위치 : %d, %d" %(n, m))  if cnd:  print(" -> 찾지 못함 ")  break |

|  |
| --- |
| 4. 제시된 조건에 알맞은 프로그램을 작성하세요. 리스트 data1 = [21, 7, 43, 65], data2 = [2, 8, 72, 52, 9]가 존재 할 때 두 리스트의 최대값, 최소값을 출력합니다.  >>>  최대값 = 72  최소값 = 2  >>>  **[소스]**  # 4  data1 = [21, 7, 43, 65]  data2 = [2, 8, 72, 52, 9]    print("최대값 = ", max(\*data1, \*data2))  print("최소값 = ", min(\*data1, \*data2)) |

|  |
| --- |
| 5. 제시된 조건에 알맞은 프로그램을 작성하세요. 1부터 100 사이의 숫자 10개를 리스트 ran\_num에 저장한 후, 저장된 수 중 5의 배수의 수만 리스트 mul\_five에 저장하고 출력하세요. (Hint: 랜덤 번호 생성 random.sample(range(1, 101), 10), numpy 이용)  >>>  ran\_num = [88 12 51 40 8 29 52 62 3 90]  mul\_five [40 90]  >>>  **[소스]**  # 5  import random  ran\_num = list(random.sample(range(1, 101), 10))  print(ran\_num)  mul\_five = []  for i in ran\_num:  if i%5 == 0:  mul\_five.append(i)    print(mul\_five) |

|  |
| --- |
| 6. 제시된 조건에 알맞은 프로그램을 작성하세요. 로또 판매장에서 랜덤 번호를 부여하는 시스템을 만들어봅니다. 로또 번호는 1~45까지의 중복되지 않는 6개 당첨 번호, 1개 보너스 번호로 구성됩니다.  >>>  로또 몇 장 드릴까요? 5  < 1 > 당첨번호 [1, 8, 12, 24, 27, 40] + 보너스 44  < 2 > 당첨번호 [9, 17, 22, 26, 29, 30] + 보너스 42  < 3 > 당첨번호 [4, 23, 27, 36, 39, 40] + 보너스 43  < 4 > 당첨번호 [5, 26, 28, 29, 33, 38] + 보너스 42  < 5 > 당첨번호 [19, 23, 27, 28, 30, 34] + 보너스 41  >>>  **[소스]**  # 6  import random  n = int(input("로또 몇 장 드릴까요? "))  rnd = 0  bonus = 0  for i in range(n):  rotto = list(random.sample(range(1, 46), 6))    while(True):  rnd = random.sample(range(1, 46), 1)  if rotto.count(rnd[0]) >= 1:  continue  else:  bonus = rnd  break  print("< {0:^2} > 당첨번호 {1} + 보너스 {2}".format((i+1), rotto, bonus[0])) |

|  |
| --- |
| 7. 제시된 조건에 알맞은 프로그램을 작성하세요. 문자열을 입력 받아 리스트 data에 입력한 문자열을 추가합니다. 빈 문자열이 입력되면 반복을 종료하고, 리스트의 전체 원소를 알파벳 순서로 출력합니다. (Hint: 빈 문자열 check : if( str=='' ) )  >>>  문자열 입력 : apple  문자열 입력 : grape  문자열 입력 : banana  문자열 입력 :  ['apple', 'banana', 'grape']  >>>  **[소스]**  # 7  myList = []  while(True):  myStr = input("문자열 입력 : ")  if myStr == "":  break    myList.append(myStr)    print(myList) |

|  |
| --- |
| 8. 제시된 조건에 알맞은 프로그램을 작성하세요. 사용자로부터 문자열을 입력 받아 각각의 문자로 분리해 리스트에 저장 한 후, 과반 수가 넘는 문자를 출력하는 프로그램을 작성합니다. 단, 입력 받은 문자열이 없으면 반복을 종료합니다.  >>>  문자열 입력 : asasass  chlist= ['a', 's', 'a', 's', 'a', 's', 's']  과반수가 넘는 문자 = s  문자열 입력 : aabb  chlist= ['a', 'a', 'b', 'b']  과반수가 넘는 문자 없음  문자열 입력 : abacadaea  chlist= ['a', 'b', 'a', 'c', 'a', 'd', 'a', 'e', 'a']  과반수가 넘는 문자 = a  문자열 입력 :  >>>  **[소스]**  # 8  while(True):  chlist = [] #입력된 문자열에서 문자 하나하나 저장하기 위한 list  clist = [] #각 문자에 따른 갯수를 저장할 list  temp = input("문자열 입력 : ")  if temp == "":  break  for i in temp:  chlist.append(i)  print("chlist = ", chlist)    chlist.sort() #서로 다른 문자 갯수를 구하기위한 정렬  word = chlist[0] #정렬 후 맨 첫번째 문자와 갯수 기록  clist.append([word, chlist.count(word)])    for i in chlist: #바뀌는 문자들도 갯수와 함께 기록  if word != i:  word = i  clist.append([word, chlist.count(word)])    print(clist) #[문자, 갯수] 리스트 출력확인  maxN = loc = -1 #과반수 확인과 그때의 문자 위치 기록  for n, v in enumerate(clist):  if maxN < v[1]:  maxN = v[1]  loc = n  elif maxN == v[1]:  loc = -1    if loc == -1:  print("과반수가 넘는 문자 없음")  elif maxN != -1:  print("과반수가 넘는 문자 = ", clist[loc][0]) |