

## ✓ AI기반 챗봇 및 OCR 개발 전문가 과정

교과목명 : m2\_programming

- 평가일 : 24.05.14
- 성명 :
- 점수 :

※ 문제당 3점으로 채점 60점 + 자유 프로그램 40점

### ✓ 1. numbers 리스트에서 15, 20, 25를 출력하세요.(for문과 if 조건 사용 필수)

numbers = [5, 15, 6, 20, 7, 25]

```
1 # 1
2 numbers = [5, 15, 6, 20, 7, 25]
3 for n in numbers:
4     if n >= 15 and n % 5 == 0:
5         print(n)
```

⇒ 15  
20  
25

```
1 # 2
2 numbers = [5, 15, 6, 20, 7, 25]
3 for n in numbers:
4     if n == 15 or n == 20 or n == 25:
5         print(n)
```

⇒ 15  
20  
25

### ✓ 2. 아래 list\_input에 대해서 주어진 사용자 함수를 일괄적으로 적용할 수 있는 2개의 내장함수를 활용하여 [1, 4, 9, 16, 25], [1, 2]를 출력하는 코드를 작성하세요.

```
1 # 사용자 함수를 선언합니다.
2 def power(item):
3     return item * item
4 def under_3(item):
5     return item < 3
6
7 # 변수를 선언합니다.
8 list_input = [1, 2, 3, 4, 5]
9
10 num_li1 = list(map(power, list_input))
11 print(num_li1)
12 num_li2 = list(filter(under_3, list_input))
13 print(num_li2)
```

⇒ [1, 4, 9, 16, 25]  
[1, 2]

### ✓ 3. 정수를 입력하면 짝수 홀수 여부를 반환해주는 프로그램을 작성하세요.(정수를 입력하지 않는 경우에 대해서 예외처리 수행)

```
1 # 예외처리 수행이 안됨.
2 i = int(input('정수를 입력해 주세요 > '))
3 if i % 2 == 0:
4     print(f"{i}는 짝수 입니다.")
5 else:
6     print(f"{i}는 홀수입니다.")
```

⇒ 정수를 입력해 주세요 > 1  
1는 홀수입니다.

```

1 # 전부다 예외처리 되버림;
2 def check_i():
3     if i % 2 == 0:
4         print(f"{i}는 짝수 입니다.")
5     else:
6         print(f"{i}는 홀수입니다.")
7
8 try:
9     i = int(input('정수를 입력해 주세요 > '))
10    result = check_i(i)
11    print()
12 except :
13    print("올바른 값을 입력하세요")

```

↻ 정수를 입력해 주세요 > 11  
올바른 값을 입력하세요

✓ 4. a의 b제곱에서 a x b를 뺀 수를 a, b가 아래와 같이 주어졌을 때 람다 함수를 이용해서 구하세요.

a = 2, b = 5

a= 5, b = 2

```

1 ans = lambda a, b: (a**b) - (a*b)
2 print(ans(2,5))
3 print(ans(5,2))

```

↻ 22  
15

✓ 5. 정수를 입력하면 천 단위로 콤마로 구분하여 출력하세요.(정수가 아닌 경우 '정수가 아닙니다.'를 출력)

예시

정수를 입력하세요: 123456789

123,456,789

```

1 i = int(input('정수를 입력해 주세요 > '))

```

✓ 6. example\_list = ["요소A", "요소B", "요소C"]를 파이썬 내장함수를 이용하여 아래와 같이 출력하세요.

0번째 요소는 요소A입니다.

1번째 요소는 요소B입니다.

2번째 요소는 요소C입니다.

```

1 example_list = ["요소A", "요소B", "요소C"]
2
3 for idx, yo in enumerate(example_list):
4     print(f'{idx}번째 요소는 {yo}입니다.')

```

↻ 0번째 요소는 요소A입니다.  
1번째 요소는 요소B입니다.  
2번째 요소는 요소C입니다.

✓ 7. 97을 40으로 나눈 몫과 나머지를 내장함수를 이용하여 구하세요.(튜플로 출력 필수)

```

1 result = divmod(97, 40)
2 print(result)

```

↻ (2, 17)

8. list\_number = [52, 273, 32, 72, 100]로 변수 선언하고 정수를 입력하면 그 정수 인덱스와 리스트에서 해당하는 값을 출력하는 프로그램을 작성하세요. 단, 모든 예외처리를 수행하며 특히 ValueError와 IndexError는 별도 구분해서 예외처리 한 후 '예외 처리 완료'를 마지막으로 출력하세요.

- 4 입력시  
정수 입력> 4  
4번째 요소: 100 예외 처리 완료
- 문자 입력시  
정수 입력> 가

정수를 입력해 주세요! 예외 처리 완료

- 5 입력시

정수 입력>5

리스트의 인덱스를 벗어났어요! 예외 처리 완료

```
1 list_number = [52, 273, 32, 72, 100]
2
3 # inumerate...?
```

✓ 9. 사용자로부터 서로 다른 숫자 두 개를 입력받고 첫 번째 와 두 번째중 큰 숫자를 구하는 프로그램을 수행하세요.

```
1 int1 = int(input('첫번째 숫자 > '))
2 int2 = int(input('두번째 숫자 > '))
3
4 if int1 > int2:
5     print(f'두 숫자중 큰 숫자는 첫번째 숫자인 {int1}입니다')
6 elif int1 < int2:
7     print(f'두 숫자중 큰 숫자는 두번째 숫자인 {int2}입니다')
8 else:
9     print('두 숫자가 같습니다.')
```

➡ 첫번째 숫자 > 12  
두번째 숫자 > 12  
두 숫자가 같습니다.

✓ 10. 1부터 100까지 아래와 같은 방식으로 2개의 수를 곱해서 가장 큰 수를 구하세요.

1 \* 99 ... 99 \* 1

```
1 max_num = 0
2 for i in range(1, 51):
3     num = i * (101 - i)
4     if num > max_num:
5         max_num = num
6
7 print("가장 큰 수:", max_num)
```

➡ 가장 큰 수: 2550

✓ 11. 현재 시간을 년-월-일 시:분:초로 출력하세요.

```
1 import datetime
2 import pytz
3
4 today = datetime.datetime.now(pytz.timezone('Asia/Seoul'))
5 print(today.strftime('%Y-%m-%d %X'))
```

➡ 2024-05-14 16:13:52

✓ 12. 1에서 백만까지 더하는데 걸리는 프로그램 실행 시간을 밀리초(ms) 단위로 구하세요.(1000 밀리초(ms) = 1초)

```
1 import time
2
3 start_time = time.time()
4 total = 0
5 for i in range(1, 1000001):
6     total += i
7 finish_time = time.time()
8
9 result_time = (finish_time - start_time) * 1000
10 print(f"프로그램 실행 시간 : {round(result_time)}ms")
```

➡ 프로그램 실행 시간 : 291ms

✓ 13. [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'a', 'b', 'c']에서 9와 'a'를 삭제 후 리스트로 출력하세요

```

1 # 1
2 li1 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'a', 'b', 'c']
3
4 li1.pop(9)
5 li1.pop(9)
6 print(li1)

```

⇒ [0, 1, 2, 3, 4, 5, 6, 7, 8, 'b', 'c']

```

1 # 2
2 li1 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'a', 'b', 'c']
3
4 li1.remove(9)
5 li1.remove('a')
6 print(li1)

```

⇒ [0, 1, 2, 3, 4, 5, 6, 7, 8, 'b', 'c']

✓ 14. 500 ~ 1000사이의 정수 중 5와 7의 공배수의 합을 구하세요

```

1 result = 0
2 for n in range(500,1001):
3     if n % 35 == 0:
4         result += n
5
6 print(f' 500 ~ 1000사이 5와 7공배수의 합 : {result}')

```

⇒ 500 ~ 1000사이 5와 7공배수의 합 : 10535

✓ 15. []을 채워서 아래와 같이 출력하세요.

```

1 numbers = [1,2,3,4,5,6,7,8,9]
2 output = [[], [], []]
3 # [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
4
5 for number in numbers:
6     output[number % 3].append(number)
7 print(output)

```

⇒ [[3, 6, 9], [1, 4, 7], [2, 5, 8]]

✓ 16. numbers 내부에 들어 있는 숫자가 몇 번 등장하는지를 출력하는 코드를 작성하세요.

답안 예시:

{3: 4, 6: 2, 5: 2, 8: 2, 2: 2, 9: 2, 7: 2, 4: 2}

```

1 numbers = [3,6,3,5,8,2,9,7,4,3,6,3,5,8,2,9,7,4]
2
3 count_dict = {}
4 for num in numbers:
5     if num in count_dict:
6         count_dict[num] += 1
7     else:
8         count_dict[num] = 1
9
10 print(count_dict)

```

⇒ {3: 4, 6: 2, 5: 2, 8: 2, 2: 2, 9: 2, 7: 2, 4: 2}

✓ 17. 클래스 Country를 완성하여 아래와 같이 출력하세요.

```

1 # 한국의 인구는 5천만명이며 수도는 서울입니다.
2 # 1인당 gnp는 USD 3만입니다.
3
4 class Country:
5     def __init__(self, contry, people, cityhall):
6         self.contry = contry
7         self.people = people
8         self.cityhall = cityhall
9     def __str__(self):
10         print(f"{self.contry}의 인구는 {self.people}천만명이며, 수도는 {self.cityhall}입니다.")
11
12
13 class Korea(Country):
14     def desc1(self, gnp):
15         self.gnp = gnp
16         print('1인당 gnp는 USD %d만입니다.' % self.gnp)
17
18 obj = Korea('한국', 5, '서울')
19 obj.__str__()
20 obj.desc1(3)

```

↻ 한국의 인구는 5천만명이며, 수도는 서울입니다.  
1인당 gnp는 USD 3만입니다.

- ✓ 18. 사칙연산을 수행하는 클래스를 작성하여 객체를 만들고 2개의 정수 10, 5를 입력한 후 연산 결과를 포맷을 이용하여 출력하세요.

```

1 class Element:
2     def __init__(self, a, b):
3         self.a = a
4         self.b = b
5
6     def add(self):
7         return self.a + self.b
8
9     def sub(self):
10         return self.a - self.b
11
12     def mult(self):
13         return self.b * self.b
14
15     def div(self):
16         if self.b == 0:
17             return "0으로 나눌 수 없습니다."
18         else:
19             return self.a / self.b
20
21 element = Element(10, 5)
22
23 print(f"덧셈 : {element.add()}")
24 print(f"뺄셈 : {element.sub()}")
25 print(f"곱셈 : {element.mult()}")
26 print(f"나눗셈 : {element.div()}")

```

↻ 덧셈 : 15  
뺄셈 : 0  
곱셈 : 25  
나눗셈 : 2.0

- ✓ 19. 올해 경과된 날짜수를 계산하여 출력하세요.

답안 예시:

오늘은 2024-01-01 이후 127일째 되는 날입니다.

```

1 import datetime
2
3 now = datetime.datetime.now()
4 gone_days = int(now.strftime('%j'))
5 print(f'오늘은 2024-01-01 이후 {gone_days-1}일째 되는 날입니다.')

```

↻ 오늘은 2024-01-01 이후 134일째 되는 날입니다.

- ✓ 20. 문자열 리스트 states를 아래 [출력]와 같이 전처리하여 출력하세요

```
states = [' Alabama ', 'Georgia!', 'Georgia', 'georgia', 'Fl0rIda', 'south carolina##', 'West virginia?']
```

[출력]

```
['Alabama',  
'Georgia',  
'Georgia',  
'Georgia',  
'Florida',  
'South Carolina',  
'West Virginia']
```

1 코딩을 시작하거나 AI로 코드를 생성하세요.

자유 프로그램 수행 결과를 실행하세요

```
1 import datetime  
2 import pytz  
3  
4 class Inventory:  
5     idx = 0  
6     threshold = 0  
7  
8     def __init__(self):  
9         self.inventory_s = {} # 재고 아이템과 수량을 저장할 딕셔너리  
10        self.inventory_p = {} # 재고 아이템과 가격을 저장할 딕셔너리  
11        self.stock_history = {}# 로그 기록을 저장할 딕셔너리  
12  
13    def add_item(self, name, stock, price):  
14        self.name = name  
15        self.stock = stock  
16        self.price = price  
17        stats = '입고'  
18        # 로그용 기록을 저장 할 메서드  
19        self.log(self.idx, self.name, self.stock, self.price, self.current_time(), stats)  
20        # 물건 재고를 재입고 시 수량은 계속 더함  
21        if name in self.inventory_s:  
22            self.inventory_s[self.name] += self.stock  
23        # 처음 물건을 넣었을 때 수량과 가격을 스톡  
24        else:  
25            self.inventory_s[self.name] = self.stock  
26            self.inventory_p[self.name] = self.price  
27  
28    def subtract_item(self, name, stock):  
29        self.name = name  
30        self.stock = stock  
31        stats = '출고'  
32        # 로그용 기록을 저장 할 메서드  
33        self.log(self.idx, self.name, self.stock, self.price, self.current_time(), stats)  
34        # 물건 재고를 출고 시 수량은 계속 빼준다  
35        if name in self.inventory_s:  
36            self.inventory_s[self.name] -= self.stock  
37        else:  
38            # 아니면 수량 그대로  
39            self.inventory_s[self.name] = self.stock  
40  
41    # 현재 입고 된 물건 이름과 수량 표시  
42    def display_stock(self):  
43        print("\n\n" )  
44        print(f"현재 {self.current_time()} 저장 된 재고량입니다 ")  
45        print('\W033[92m' + "===== " + '\W033[0m')  
46        for n, s in self.inventory_s.items():  
47            print(f'{n}, {s}개')  
48            # 재고 부족 메세지  
49            if s < self.threshold:  
50                print(f'\W033[91m' + '경고!' + '\W033[0m', end='')  
51                print(f' : {n} 재고가 부족합니다. 해당 물건을 더 채워주세요')  
52        print('\W033[92m' + "===== " + '\W033[0m')  
53  
54    # 현재 입고 된 물건 가격 표시  
55    def display_price(self):  
56        print("\n\n" )  
57        print(f"현재 {self.current_time()} 저장 된 재고의 가격입니다 ")  
58        print('\W033[95m' + "===== " + '\W033[0m')  
59        for n, p in self.inventory_p.items():  
60            print(f'{n:<10} : {p}원')  
61        print('\W033[95m' + "===== " + '\W033[0m')  
62  
63    def display_log(self):  
64        print('\W033[94m' + "===== " + '\W033[0m')  
65        for name, stock, price, today, stats in self.stock_history.values():
```



```

143         print('재고가 없습니다. 재고를 확인해주세요.')
144         continue
145     else:
146         break
147     if name in inventory.inventory_s:
148         # 재고가 있는지 확인
149         while "the answer is invalid":
150             # 재고 있으면 가격 변경 제시
151             price_change = str(input(f'{name}의 가격을 바꾸시겠습니까? (y / n) : ')).lower().strip()
152             if price_change[0] == 'y':
153                 price = int(input('가격 : '))
154                 inventory.inventory_p[name] = price
155                 inventory.add_item(name, stock, price)
156                 print("해당 물건이 입고되었습니다.")
157                 break
158             # 가격 변경 없으면 그 가격으로
159             elif price_change[0] == 'n':
160                 price = inventory.inventory_p[name]
161                 inventory.add_item(name, stock, price)
162                 print("해당 물건이 입고되었습니다.")
163                 break
164             else:
165                 print('y or n 만 입력해주세요')
166         # 새로운 물건 일 때
167     else:
168         try:
169             price = int(input('가격 : '))
170         except:
171             print('숫자만 넣어주세요')
172             continue
173         else:
174             inventory.add_item(name, stock, price)
175             print("해당 물건이 입고되었습니다.")
176
177 # 기능 2번 출고
178 elif choice == '2':
179     # 물건이 없을 때 출력
180     if inventory.inventory_s == {}:
181         print("입고된 물건이 없습니다.")
182
183     # 물건이 있을 시
184     else:
185         name = input('물건이름 : ')
186         # 재고가 있는지 확인
187         if name not in inventory.inventory_s:
188             print("해당 물건이 없습니다. 재고를 확인해주세요")
189         else:
190             # 재고 있으면 출고
191             while True:
192                 try:
193                     stock = int(input('수량 : '))
194                 except:
195                     print('숫자만 넣어주세요')
196                     continue
197                 else:
198                     if inventory.inventory_s[name] >= stock:
199                         inventory.subtract_item(name, stock)
200                         print("해당 물건이 출고되었습니다.")
201                         break
202                     # 출고 할 재고가 현재 적재된 물건보다 많으면
203                     else:
204                         print("재고가 부족합니다.")
205                         break
206
207 # 기능 3번 현재 입고 된 물품 가격
208 elif choice == '3':
209     inventory.display_price()
210
211 # 기능 4번 현재까지 입/출고 수정 된 물품들 로그
212 elif choice == '4':
213     inventory.display_log()
214
215 # 기능 5번 재고에 있는 물품 삭제
216 elif choice == '5':
217     name = input('삭제 할 물건이름을 적어주세요 : ')
218     inventory.delete_item(name)
219
220 # 기능 6번 프로그램 종료
221 elif choice == '6':
222     print("프로그램을 종료합니다.")
223     break
224
225 else:
226     print('잘못된 입력입니다. 다시 입력해주세요')

```



```
231         print('0 ~ 6까지 숫자만 넣어주세요')
232
233 __main__ = Inventory()
234 __main__.play_code()
```



현재 05/14 PM 04:32:09 저장 된 재고량입니다

감자깡, 10개  
고구마깡, 10개

사용하실 기능 번호를 넣어주세요 1. 입고 / 2. 출고 / 3. 가격확인 / 4. 로그확인 / 5. 물건삭제 / 6. 프로그램 종료 : 2  
물건이름 : 감자깡  
수량 : 6  
해당 물건이 출고되었습니다.

현재 05/14 PM 04:32:20 저장 된 재고량입니다

감자깡, 4개  
**경고!** : 감자깡 재고가 부족합니다. 해당 물건을 더 채워주세요  
고구마깡, 10개

사용하실 기능 번호를 넣어주세요 1. 입고 / 2. 출고 / 3. 가격확인 / 4. 로그확인 / 5. 물건삭제 / 6. 프로그램 종료 : 3

현재 05/14 PM 04:32:26 저장 된 재고의 가격입니다

감자깡 : 1500원  
고구마깡 : 1200원

현재 05/14 PM 04:32:26 저장 된 재고량입니다

감자깡, 4개  
**경고!** : 감자깡 재고가 부족합니다. 해당 물건을 더 채워주세요  
고구마깡, 10개

사용하실 기능 번호를 넣어주세요 1. 입고 / 2. 출고 / 3. 가격확인 / 4. 로그확인 / 5. 물건삭제 / 6. 프로그램 종료 : 5  
삭제 할 물건이름을 적어주세요 : 감자깡  
해당 물건이 삭제되었습니다.

현재 05/14 PM 04:32:40 저장 된 재고량입니다

고구마깡, 10개

사용하실 기능 번호를 넣어주세요 1. 입고 / 2. 출고 / 3. 가격확인 / 4. 로그확인 / 5. 물건삭제 / 6. 프로그램 종료 : 4

감자깡 : 10개, 1500원, 05/14 PM 04:32:00, 입고  
고구마깡 : 10개, 1200원, 05/14 PM 04:32:09, 입고  
감자깡 : 6개, 1200원, 05/14 PM 04:32:20, 출고  
감자깡 : 4개, 1500원, 05/14 PM 04:32:40, 삭제

현재 05/14 PM 04:32:44 저장 된 재고량입니다

고구마깡, 10개