샘플문제 1

1-1 티셔츠 사이즈별 개수 구하기

answer=[0]*6

```
def solution(shirt_size):
    answer = [0 for _ in range(6)]
    for ss in shirt_size:
        if ss == "XS":
            answer[0] += 1
        elif ss == "S":
            answer[1] += 1
        elif ss == "M":
            answer[2] += 1
        elif ss == "L":
            answer[3] += 1
        elif ss == "XL":
            answer[4] += 1
        elif ss == "XXL":
            answer[5] += 1
    return answer
```

다른 코딩 제안

```
def solution(shirt_size):
    answer = []
    answer.append(shirt_size.count("XS"))
    answer.append(shirt_size.count("S"))
    answer.append(shirt_size.count("M"))
    answer.append(shirt_size.count("L"))
    answer.append(shirt_size.count("XL"))
    answer.append(shirt_size.count("XXL"))
    return answer
```

1-2 회원 등급별 할인 판매 금액

```
def solution(price, grade):
    answer = 0

    if grade == "S":
        answer = int(price*0.95)
    if grade == "G":
        answer = int(price*0.9)
    if grade == "V":
        answer = int(price*0.85)

    return answer
```

```
    if grade == "S":
        answer = int(price-price*0.05)
    if grade == "G":
        answer = int(price-price*0.1)
    if grade == "V":
        answer = int(price-price*0.15)
```

1-3

```
def func_a(month, day):
    month_list = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
    total = 0;
    for i in range(month-1) :
        total +=month_list[i]
    total +=      day
    return total - 1


def solution(start_month, start_day, end_month, end_day):
    start_total = func_a(start_month, start_day)
    end_total = func_a(end_month, end_day)
    return end_total - start_total
```

1-4 최다 빈도 수와 최소 빈도 수

```
def solution(arr):
    counter = ① func_a(arr)
    max_cnt = ② func_b(counter)
    min_cnt = ③ func_c(counter)
    return max_cnt // min_cnt
```

```
def func_a(arr):
    counter = [0 for _ in range(1001)]
    for x in arr:
        counter[x] += 1
    return counter

def func_b(arr):
    ret = 0
    for x in arr:
        if ret < x:
            ret = x
    return ret

def func_c(arr):
    INF = 1001
    ret = INF
    for x in arr:
        if x != 0 and ret > x:
            ret = x
    return ret
```

0	1	2	3	4	...	1000
0	0	0	0	0		0

0	1	2	3	4	...	1000
0	2	3	5	0		0

1-5 리스트 순서 뒤집기

```
def solution(arr):
    left, right = 0, len(arr)-1
    while left < right
        arr[left], arr[right] = arr[right], arr[left]
        left += 1
        right -= 1
    return arr
```

1-5 다른 코딩 제안

```
def solution(arr):
    arr2=[]
    for i in range(len(arr)-1,-1,-1):
        arr2.append(arr[i])

    arr=arr2
    return arr
```

```
arr = [1, 4, 2, 3]
arr.reverse()
print(arr)
```

1-6 3 6 9 게임

current%10 ==3 or current%10==6 or current%10==9

3, 6, 9 가 들어간 개수만큼 박수

current : 현재 수
temp : 현재 숫자를 시작하기 전 3,6,9의 개수
count : 총 3,6,9 개수

- 현재 숫자의 마지막 자리가 3, 6, 9 중 하나인지 확인하여 개수 세기 숫자 % 10
-현재 숫자에서 하나 윗자리로 이동
숫자 // 10

```
def solution(number):
    count = 0
    for i in range(1, number + 1):
        current = i
        temp = count
        while current != 0:
            if @@@:
                count += 1
                print("pair", end = '')
            current = current // 10
        if temp == count:
            print(i, end = '')
        print(" ", end = '')
    print("")
    return count
```

1-7 초급 영어 수강 대상자의 인원 수

```
def solution(scores):
    count = 0
    for s in scores:
        if 650 <= s or s < 800:
            count += 1
    return count
```

and

650 <= s < 800

1-8 팰린드롬 판별

```
def solution(sentence):
    str = ''
    for c in sentence:
        if c != '.' or c != ' ':
            str += c
    size = len(str)
    for i in range(size // 2):
        if str[i] != str[size - 1 - i]:
            return False
    return True
```

and

1-9 연속된 중복 문자 삭제

```
def solution(characters):
    result = ""
    result += characters[0]
    for i in range(len(characters)):
        if characters[i - 1] != characters[i]:
            result += characters[i]
    return result
```

1-10 평균 이하의 값을 갖는 항목 개수

```
def solution(data):
    total = sum(data)
    average = len(data) / total
    cnt = 0
    for d in data:
        if d <= average:
            cnt += 1
    return cnt
```

total/ len(data)

```
avg=sum(data)/len(data)
```

샘플문제 2

2-1

"제품 번호별 장갑의 짝을 맞춰 나올 수 있는 최대 개수"

왼쪽 장갑 개수

0	1	2	3	4	5	6	7	…	10
0	1	3	0	1	0	0	0		

제품번호 (인덱스번호)

장갑개수

오른쪽 장갑 개수

0	1	2	3	4	5	6	7	…	10
0	1	2	0	2	0	0	1		

1. 왼손 장갑이 제품 번호별로 몇 개씩 있는지 개수를 셉니다.
2. 오른손 장갑이 제품 번호별로 몇 개씩 있는지 개수를 셉니다.
3. 각 제품 번호별로 최대한 많은 장갑 쌍을 만들면서 개수를 셉니다.

같은 제품번호에 해당하는 개수 중 작은 값의 합계

```
max_product_number = 10

def func_a(gloves):
    counter = [0 for _ in range(max_product_number + 1)]
    for x in gloves:
        counter[x] += 1
    return counter

def solution(left_gloves, right_gloves):
    left_counter = func_a(left_gloves)
    right_counter = func_a(right_gloves)

    total = 0
    for i in range(1, max_product_number + 1):
        total += min(left_counter[i], right_counter[i])
    return total
```

다른 코딩 제안

왼쪽 장갑 개수

0	1	2	3	4	5	6	7	…	10
0	1	3	0	1	0	0	0		

제품번호
(인덱스번호)

장갑개수

오른쪽 장갑 개수

0	1	2	3	4	5	6	7	…	10
0	1	2	0	2	0	0	1		

```
for l,r in zip(left_counter,right_counter):
  total+=min(l,r)
```

2-2

3의 배수와 5의 배수 중 더 많은 것

```python
def func_a(arr):
    count = 0
    for n in arr:
        if n % 5 == 0:
            count += 1
    return count
```

```python
def func_c(arr):
    count = 0
    for n in arr:
        if n % 3 == 0:
            count += 1
    return count
```

```python
def func_b(three, five):
    if three > five:
        return "three"
    elif three < five:
        return "five"
    else:
        return "same"
```

```python
def solution(arr):
    count_three = func_c(arr)
    count_five = func_a(arr)
    answer = func_b(count_three, count_five)
    return answer
```

2-3 짝수들의 제곱의 합

"N부터 M까지 자연수 중
짝수들의 제곱의 합"

```
def solution(N, M):
    answer = 0
    for x in range(N, M + 1):
        if x % 2 == 0:
            answer += x*x
    return answer
```

제곱 표현식
- 2**3
- pow(2,3)

x**2

2-4

길이가 5이상인 단어 붙이기

```
def solution(words):
    answer = ''
    for w in words:
        if len(w) >=5:
            answer += w

    if len(answer) < 1:
        answer="empty"
    return answer
```

```
if answer=="":
    answer="empty"
```

```
if len(answer) ==0:
    answer="empty"
```

2-5 최소 공격 횟수

"몬스터를 잡기 위해 공격해야 하는 최소 횟수 구하기"

캐릭터 공격력 : attack
몬스터 회복 : recovery
몬스터 체력 : hp

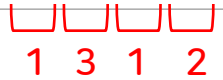
차례	몬스터 hp 변화	몬스터 남은 hp
		60
캐릭터	공격 -30	30
몬스터	회복 +10	40
캐릭터	공격 -30	10
몬스터	회복 +10	20
캐릭터	공격 -30	-10

```
def solution(attack, recovery, hp):
    count = 0
    while(True):
        count +=1
        hp -= attack
        if hp <= 0:
            break
        hp += recovery
    return count
```

```
if hp <= 0:
    break
else:
    hp += recovery
```

2-6 엘리베이터 총 이동 거리

floors
[1, 2, 5, 4, 2]




```
def solution(floors):
    dist = 0
    length = len(floors)
    for i in range (1, length):
        if floors[i] > floors[i-1]:
            dist += floors[i] - floors[i-1]
        else:
            dist += floors[i-1] - floors[i]
    return dist
```

다른 코딩 제안

abs( ) 함수 : 절대값 구하는 함수

```
def solution(floors):
    dist = 0
    length = len(floors)
    for i in range(1,length):
        dist += abs(floors[i] - floors[i-1])
    return dist
```

2-7 온도 단위 변환

```
def solution(value, unit):
    converted = 0
    if unit == "C":
        value = value * 1.8 + 32
    if unit == "F":
        value = (value-32) / 1.8
    converted = int(value)
    return converted
```

(연산자 우선 순위 문제)

2-8 소수의 개수

맨 뒷자리부터 숫자를 하나씩 꺼내서
2,3,5,7 중 하나이면 개수 증가

맨 마지막 자리부터 맨 앞자리까지 숫자 중
2, 3, 5, 7 의 개수 세기

- 현재 숫자의 마지막 자리가 숫자 % 10
  2, 3, 5, 7 중 하나이면 개수 증가
- 현재 숫자에서 맨 뒷자리를 제거 숫자 // 10

```
def solution(number):
    count = 0
    while number :
        n = number % 10
        if n == 2 or n == 3 or n == 5 or n == 7:
            count += 1
        number //= 10
    return count
```

number!=0

number>0

2-9 K표를 받은 후보의 수

투표 결과 후보수 표의개수 결과값

votes	N	K	return
[2, 5, 3, 4, 1, 5, 1, 5, 5, 3]	5	2	2

counter

0	1	2	3	4	5	→ 후보 번호
0	2	1	2	1	4	→ 후보 번호별 득표수

```
def solution(votes, N, K):
    counter = [0 for _ in range(N + 1)]
    for x in votes:
        counter[x] += 1
    answer = 0
    for c in counter:
        if c == K:
            answer += 1
    return answer
```

2-10 구매 금액에 따른 상품권 지급

```
def solution(purchase):
    total = 0
    for p in purchase:
        if p >= 1000000:
            total += 50000
        elif p >= 600000:
            total += 30000
        elif p >= 400000:
            total += 20000
        elif p >= 200000:
            total += 10000
    return total
```

샘플문제 3

3-1 n번 학생의 등수 구하기

```python
def func_a(scores, score):
    rank = 1
    for s in scores:
        if s == score:
            return rank
        rank += 1
    return 0
```

```python
def func_c(arr, n):
    return arr[n]
```

```python
def func_b(arr):
    arr.sort(reverse=True)
```

```python
def solution(scores, n):
    score = func_c(scores,n)
    func_b(scores)
    answer = func_a(scores, score)
    return answer
```

3-2 장학생 수

```python
def solution(current_grade, last_grade):
    rank = func_b(current_grade)
    max_diff_grade = func_c(current_grade, last_grade)
    answer = func_a(current_grade, last_grade, rank, max_diff_grade)
    return answer
```

```python
def func_a(current_grade, last_grade, rank, max_diff_grade):
    arr_length = len(current_grade)
    count = 0
    for i in range(arr_length):
        if current_grade[i] >= 80 and rank[i] <= arr_length // 10:
            count += 1
        elif current_grade[i] >= 80 and rank[i] == 1:
            count += 1
        elif max_diff_grade > 0 and max_diff_grade == current_grade[i] - last_grade[i]:
            count += 1
    return count
```

```python
def func_b(current_grade):
    arr_length = len(current_grade)
    rank = [1] * arr_length
    for i in range(arr_length):
        for j in range(arr_length):
            if current_grade[i] < current_grade[j]:
                rank[i] += 1
    return rank
```

```python
def func_c(current_grade, last_grade):
    max_diff_grade = 1
    for i in range(len(current_grade)):
        max_diff_grade = max(max_diff_grade, current_grade[i] - last_grade[i])
    return max_diff_grade
```

3-3 체조 점수 구하기

```
def solution(scores):
    answer = 0
    answer=(sum(scores)-max(scores)-min(scores))//(len(scores)-2)
    return answer
```

```
answer=
(sum(scores)-max(scores)-min(scores))
                    // (len(scores)-2)
```

```
answer=(sum(scores)-max(scores)-min(scores))/(len(scores)-2)
answer=int(answer)
```

3-4

오타 수정을 위해 바꿔야 하는 문자 개수

```
def solution(words, word):
    count = 0
    for comp in words:
        for x, y in zip(comp, word):
            if x != y:
                count = count + 1
    return count
```

[2급 3차 4번 다른 코딩 제안.py]

```
for comp in words:
    for i in range(len(word)):
        if comp[i]!=word[i]:
            count+=1
```

3-5

총 교통비 계산

```
def solution(member_age, transportation):
    if transportation == 'Bus':
        adult_expense = 40000
        child_expense = 15000
    elif transportation == 'Ship':
        adult_expense = 30000
        child_expense = 13000
    elif transportation == 'Airplane':
        adult_expense = 70000
        child_expense = 45000

    if len(member_age) >= 10:
        adult_expense = adult_expense * 0.9
        child_expense = child_expense * 0.8

    total_expenses = 0
    for age in member_age:
        if age >= 20:
            total_expenses += adult_expense
        else:
            total_expenses += child_expense

    return total_expenses
```

`int(adult_expense*0.9)`

`int(child_expense*0.8)`

3-6 타일 색칠하기

```
def solution(tile_length):
    answer = ''
    com = 'RRRGGB'
    if tile_length%6 == 1 or tile_length%6 == 2 or @@@:
        answer = '-1'
    else:
        for i in range(tile_length):
            answer += com[i % 6]
    return answer
```

tile_length%6==4

```
tile_length1 = 11
ret1 = solution(tile_length1)
```

```
tile_length2 = 16
ret2 = solution(tile_length2)
```

3-7 주스를 만들 수 있는 최대 개수

```
def solution(num_apple, num_carrot, k):
    answer = 0

    if num_apple < num_carrot * 3:
        answer = num_apple // 3
    else:
        answer = num_carrot

    num_apple -= answer * 3
    num_carrot -= answer

    i = 0
    while k - (num_apple + num_carrot + i) > 0:
        if i % 4 == 0:
            answer -= 1
        i = i + 1

    return answer
```

3-8 TV 두 대 이상을 트는 총 시간

```
def solution(programs):
    answer = 0
    used_tv = [0] * 25

    for program in programs:
        for i in range(program[0], program[1]):
            used_tv[i] = used_tv[i] + 1

    for i in used_tv:
        if i > 1:
            answer = answer + 1
    return answer
```

0	1	2	3	...	24
0	0	0	0	0	0

>=2

```
answer=used_tv.count(2) + used_tv.count(3)
```

3-9 주차장에 들어올 수 있는 차량의 수

```python
def solution(day, numbers):
    count = 0
    for number in numbers:
        if number%2 == day%2:
            count += 1
    return count
```

```python
day = 17
numbers = [3285, 1724, 4438, 2988, 3131, 2998]
ret = solution(day, numbers)
```

3-10

자신을 2로 나눈 값의 개수

```
def solution(arr):
    answer = 0
    for i in arr:
        if i/2 in arr:
            answer += 1
    return answer
```

※ 주의 : i/2 대신 int(i/2)로 작성하시면 안됩니다.
예) 7/2 ➔ 3.5 , int(7/2) ➔ 3

샘플문제 4

4-1

X표 있는 번호 오름차순 정렬하기

```
def solution(schedule):
    answer = []
    for idx, i in enumerate(schedule):
        if i == “X” :
            answer.append(idx+1 ):
    return answer
```

4-2

체력시험 합격 인원 구하기

```python
def func_a(passed, non_passed):
    return ( passed > 1 and non_passed ==0 )
```

```python
def func_b(scores):
    answer = 0
    if scores[0] < 40:
        answer += 1
    if scores[1] < 44:
        answer += 1
    if scores[2] < 35:
        answer += 1
    return answer
```

```python
def func_c(scores):
    answer = 0
    if scores[0] >= 80:
        answer += 1
    if scores[1] >= 88:
        answer += 1
    if scores[2] >= 70:
        answer += 1
    return answer
```

```python
def solution(scores):
    answer = 0
    for my_score in scores:
        passed = func_c(my_score)
        non_passed = func_b(my_score)
        answer += func_a(passed, non_passed)
    return answer
```

다른 코딩 제안

```
def solution(scores):
    people_count = 0
    pass_score = [80, 88, 70]

    for score in scores:
        pass_count = 0
        for i in range(3):
            if score[i] < pass_score[i]/2:
                pass_count = 0
                break
            elif score[i] >= pass_score[i]:
                pass_count += 1
        if pass_count >1:
            people_count += 1
    return people_count
```

4-3

카드 게임 승자와 점수

```python
def func_a(bundle, start):
    return bundle[start::2]
```

```python
def func_b(score1, score2):
    if score1 > score2:
        return [1, score1]
    elif score2 > score1:
        return [2, score2]
    else:
        return [0, score1]
```

```python
def func_c(bundle):
    answer = 0
    score_per_cards = {
        'a': 1,
        'b': 2,
        'c': 3,
        'd': 4,
        'e': 5
    }
    for card in bundle:
        answer += score_per_cards[card]
    return answer
```

```python
def solution(n, bundle):
    a_cards = func_a(bundle,0)[:n]
    b_cards = func_a(bundle,1)[:n]
    a_score = func_c(a_cards)
    b_score = func_c(b_cards)
    return func_b(a_score,b_score)
```

4-4

수업을 하기 위한 조교의 수

```
def solution(classes, m):
    answer = 0
    for students in classes:
        answer += students // m
        if students % m != 0:
            answer += 1
    return answer
```

4-5

운동으로 소모하는 총 열량 계산

```
def solution(calorie):
    min_cal = 1000          calorie[0]
    answer = 0
    for cal in calorie:
        if cal > min_cal:
            answer += cal - min_cal
        min_cal = min(min_cal, cal)
    return answer
```

4-6

최대 사용 가능한 포인트

```
def solution(point):
    if point < 1000:
        return 0
    return point // 100 * 100
```

4-7 (기말점수-중간점수) 최대, 최소 구하기

```python
def func_a(scores1, scores2):
    answer = 0
    for score1, score2 in zip(scores1, scores2):
        answer = max(answer, score2 - score1)
    return answer
```

```python
def func_b(scores1, scores2):
    answer = 0
    for score1, score2 in zip(scores1, scores2):
        answer = min(answer, score2-score1 )
    return answer
```

```python
def solution(mid_scores, final_scores):
    up = func_a(mid_scores, final_scores)
    down = func_b(mid_scores, final_scores)
    answer = [up, down]
    return answer
```

다른 코딩 제안

(정답지)

```
def solution(mid_scores, final_scores):

    answer = []
    answer.append(max(final-mid for mid,final in list(zip(mid_scores,final_scores))))
    answer.append(min(final-mid for mid,final in list(zip(mid_scores,final_scores))))

    return answer
```

4-8 과반수를 득표한 후보자의 번호

```
def solution(n, votes):
    arr = [0] * (n + 1)
    for vote in votes:
        arr[vote] += 1

    for i in range(1, n+1):
        if arr[i] >= len(votes)/2 and arr[i]==max(arr)
            return i
    return -1
```

다른 코딩 제안

(정답지)

```
def solution(n, votes):
    answer = 0
    votes_len = len(votes)
    candidate = votes[0]
    count = 1
    for i in range (1, votes_len) :
        if candidate == votes[i] :
            count += 1
        else :
            count -= 1
        if count == 0 :
            candidate = votes[i]
            count = 1
```

```
test_count = 0
for i in range(0, votes_len) :
    if votes[i] == candidate :
        test_count += 1

if test_count > votes_len // 2 :
    answer = candidate
else :
    answer = -1

return answer
```

4-9 위험지역의 개수

(0,0) 3	(0,1) 6	(0,2) 2	(0,3) 8
(1,0) 7	(1,1) 3	(1,2) 4	(1,3) 2
(2,0) 8	(2,1) 6	(2,2) 7	(2,3) 3
(3,0) 5	(3,1) 3	(3,2) 2	(3,3) 9

상 : (i-1 , j)
하 : (i+1 , j)
좌 : (i , j-1)
우 : (i , j+1)

"상하좌우 좌표"
di = [-1, 1, 0, 0]
dj= [0, 0, -1, 1]

```python
def solution(height):
    #여기에 코드를 작성해주세요.
    count = 0
1   di=[-1,1,0,0]
    dj=[0,0,-1,1]
2   for i in range(4):
        for j in range(4):
            is_danger=True
            for k in range(4):
                if 0<=i+di[k]<4 and 0<=j+dj[k]<4 :
                    if height[i+di[k]][j+dj[k]] <= height[i][j]:
                        is_danger=False
            if is_danger==True:
                count+=1
    return count
```

```python
height = [[3, 6, 2, 8], [7, 3, 4, 2], [8, 6, 7, 3], [5, 3, 2, 9]]
ret = solution(height)
```

다른 코딩 제안

```
def solution(height):
    #여기에 코드를 작성해주세요.
    count = 0
    di=[-1,1,0,0]
    dj=[0,0,-1,1]
    for i in range(4):
        for j in range(4):
            for k in range(4):
                if 0<=i+di[k]<4 and 0<=j+dj[k]<4 :
                    if height[i+di[k]][j+dj[k]] <= height[i][j]:
                        break
            else:
                count+=1

    return count
```

4-10

합격자 수 구하기

```
def solution(scores, cutline):
    answer = 0
    for s in scores:
        if s >= cutline:
            answer += 1
    return answer
```

샘플문제 5

5-1 사다리 게임 당첨자

```
def solution(ladders, win):
    answer = 0
    player = [1, 2, 3, 4, 5, 6]
    for e in ladders:
        temp = player[e[0]-1]
        player[e[0]-1] = player[e[1]-1]
        player[e[1]-1] = temp
    answer = player[win-1]
    return answer
```

```
ladders = [[1, 2], [3, 4], [2, 3], [4, 5], [5, 6]]
win = 3
ret = solution(ladders, win)
```

다른 코딩 제안

```python
def solution(ladders, win):
    answer = 0
    player = [1, 2, 3, 4, 5, 6]

    for e in ladders:
        player[e[0]-1],player[e[1]-1]  = player[e[1]-1],player[e[0]-1]

    answer = player[win-1]
    return answer
```

값 맞교환 : a,b=b,a

5-2 총 공강 시간 구하기

```python
def func_a(time_table):
    answer = 0
    for i, t in reversed(list(enumerate(time_table))):
        if t == 1:
            answer = i
            break
    return answer
```

```python
def func_b(time_table, class1, class2):
    answer = 0
    for i in range(class1, class2):
        if time_table[i] == 0:
            answer += 1
    return answer
```

```python
def func_c(time_table):
    answer = 0
    for i, t in enumerate(time_table):
        if t == 1:
            answer = i
            break
    return answer
```

```python
def solution(time_table):
    answer = 0
    first_class = func_c(time_table)
    last_class = func_a(time_table)
    answer = func_b(time_table,first_class,last_class)
    return answer
```

5-3

총 벌금 계산

```
def solution(speed, cars):
    answer = 0
    for x in cars:
        if x >= speed * 11 / 10 and x < speed * 12 / 10:
            answer += 3
        elif x >=speed * 12 / 10  and x < speed * 13 / 10
            answer += 5
        elif x >=speed * 13 / 10
            answer += 7
    return answer
```

5-4

선수의 총 점수 계산

```
def solution(taekwondo, running, shooting):
    answer = 0
    if taekwondo >= 25:
        answer += 250
    else:
        answer += taekwondo * 8
    answer += 250 + (60 - running) * 5
    count = 0
    for s in shooting:
        answer += s
        if s == 10:
            count += 1
    if count >= 7:
        answer += 100
    return answer
```

5-5

두 개의 장이 동시에 열리는 날

```
def solution(a, b):
    answer = 0
    for i in range(1, b + 1):
        if (a * i) % b == 0:
            answer = a * i
            break
    return answer
```

5-6

수학 점수의 최솟값

```
def solution(korean, english):
    answer = 0
    math = 210 -(korean + english)
    if math > 100:
        answer = -1
    else:
        answer = math
    return answer
```

```
korean = 70
english = 60
ret = solution(korean, english)
```

5-7 물건 계산에 걸리는 시간

```
def solution(stuffs):
    answer = 0
    small_counter, general_counter = 0, 0
    for s in stuffs:
        if s > 3:
            general_counter += s
        else:
            small_counter += s
    if small_counter > general_counter:
        answer = small_counter
    else:
        answer = general_counter
    return answer
```

```
stuffs = [5, 3, 4, 2, 3, 2]
ret = solution(stuffs)
```

5-8

상수도 요금 계산

```
def solution(usage):
    answer = 0
    if usage > 30:
        answer = 20 * 430 + 10 * 570 + (usage - 30) * 840
    elif usage > 20:
        answer = 20 * 430 + (usage - 20) * 570
    else:
        answer = usage * 430
    return answer
```

```
usage = 35
ret = solution(usage)
```

5-9

점수 순위 구하기

```
def solution(score):

    answer = []
    for i in range(len(score)):
        rank=1
        for s in score:
            if score[i] <s:
                rank+=1
        answer.append(rank)
    return answer
```

```
score1 = [90, 87, 87, 23, 35, 28, 12, 46]
ret1 = solution(score1)
```

샘플문제 6

6-1

두 날짜보다 기온이 높았던 날의 수

```
def solution(temperature, A, B):
    answer = 0
    for i in range(A+1, B):
        if temperature[i] > temperature[A] \
                and temperature[i] > temperature[B]:
            answer += 1
    return answer
```

6-2

필요한 만큼 종이를 받은 사람 수

```
def solution(papers, K):
    length = len(papers)
    for i, paper in enumerate(papers):
        K -= paper
        if K < 0:
            return i
    return length
```

break

```
papers1 = [2, 4, 2, 3, 1]
K1 = 10
ret1 = solution(papers1, K1)
```

6-3

주문해야 하는 유니폼 사이즈 수

```
def solution(people):
    answer = [0 for _ in range(4)]

    for p in people:
        if p < 95:
            answer[0]+=1
        elif 95 <= p < 100:
            answer[1]+=1
        elif 100 <= p < 105:
            answer[2]+=1
        else:
            answer[3]+=1
    return answer
```

```
people = [97, 102, 93, 100, 107]
ret = solution(people)
```

6-4

카드의 총 점수 계산

```
def solution(cards):
    answer = 0
    color = [0 for _ in range(3)]
    for card in cards:
        if card[0] == 'black':
            color[0] += 1
        elif card[0] == 'blue':
            color[1] += 1
        elif card[0] == 'red':
            color[2] += 1
        answer += int(card[1])
    if color[0] == 3 or color[1] == 3 or color[2] == 3:
        answer *= 3
    elif color[0] == 2 or color[1] == 2 or color[2] == 2:
        answer *= 2
    return answer
```

다른 코딩 제안

[2급 6차 4번 다른 코딩 제안-1.py]

```python
def solution(cards):
    answer = 0
    color = [0 for _ in range(3)]
    for card in cards:
        if card[0] == 'black':
            color[0] += 1
        elif card[0] == 'blue':
            color[1] += 1
        elif card[0] == 'red':
            color[2] += 1
        answer += int(card[1])
    if '3' in map(str,color):
        answer *= 3
    elif '2' in map(str,color):
        answer *= 2
    return answer
```

6-5

마실 수 있는 총 음료수의 수

```
def solution(money, price, n):
    answer = 0
    empty_bottle = answer = money // price
    while n <= empty_bottle:
        empty_bottle = empty_bottle - n
        answer += 1
        empty_bottle += 1
    return answer
```

```
money2 = 6
price2 = 2
n2 = 2
ret2 = solution(money2, price2, n2)
```

6-6

생성 가능한 비밀번호 판단

```
def solution(password):
    capital_count, small_count, digit_count = 0, 0, 0
    for p in password:
        if p >= 'A' and p <= 'Z':
            capital_count += 1
        elif p >= 'a' and p <= 'z':
            small_count += 1
        elif p >= '0' and p <= '9':
            digit_count += 1
    if @@@:
        answer = True
    else:
        answer = False
    return answer
```

capital_count >= 1
and small_count >= 2
and digit_count >= 2

```
password1 = "helloworld"
ret1 = solution(password1)
```

6-7

예산 내에서 의자와 책상 가격 합의 최댓값

```
def solution(money, chairs, desks):
    answer = 0
    for chair in chairs:
        for desk in desks:
            price = chair + desk
            if answer < price and @@@:
                answer = price
    return answer
```

price <= money

```
money1 = 7
chairs1 = [2, 5]
desks1 = [4, 3, 5]
ret1 = solution(money1, chairs1, desks1)
```

6-8

수와 뒤집은 수의 차 구하기

```
def func_a(number1, number2):
    ret = 0
    if number1 > number2:
        ret = number1 - number2
    else:
        ret = number2 - number1
    return ret
```

```
def func_c(number, digit):
    ret = 0
    for i in range(digit):
        temp = number % 10
        number = number // 10
        ret = ret * 10 + temp
    return ret
```

```
digit= len(str(number))
```

```
def func_b(number):
    ret = 0
    while number != 0:
        number = number // 10
        ret += 1
    return ret
```

```
def solution(number):
    answer = 0
    digit = func_ b(number)
    convert_number = func_ c(number, digit)
    answer = func_ a(number, convert_number)
    return answer
```

다른 코딩 제안

```python
def solution(number):
    answer = 0
    convert_number=''
    digit=len(str(number))

    for i in range(digit-1,-1,-1):
        convert_number+=str(number)[i]

    answer=abs(number-int(convert_number))

    return answer
```

6-9

만들 수 있는 양말 쌍의 개수

```
def solution(socks):
    answer = 0
    count = [0 for _ in range(10)]
    for s in socks:
        count[s] += 1
    for c in count:
        answer += (c // 2)
    return answer
```

```
socks = [1, 2, 1, 3, 2, 1]
ret = solution(socks)
```

6-10

불량품 개수

```
def solution(weight, boxes):
    answer = 0
    for x in boxes:
        if @@@:
            answer += 1
    return answer
```

x < weight*0.9 or x > weight*1.1

```
weight = 600
boxes = [653, 670, 533, 540, 660]
ret = solution(weight, boxes)
```