문제 8번

```python
# 함수 내부에 불필요한 print문이 있는 경우 오답으로 처리가 됩니다.
def caesar(word, n):
    word_list = [] # 입력받은 단어를 문자로 분해하여 저장할 리스트 생성
    word_list_code = [] # 분해한 문자를 아스키 코드로 변환하여 저장할 리스트 생성
    word_list_chr = [] # 시저 암호로 변환한 아스키 코드를 다시 문자로 변환하여 저장할 리스트 생성
    for i in word: # 입력받은 word 문자열을 순회하면서
        word list.append(ord(i)) # 문자열의 각 문자들을 word list 리스트에 아스키 코드로 변환하여 추가
    for code in word_list: # 아스키 코드가 담긴 word_list를 순회하면서
        if code <= 90 and code >= 65:
            aa = code + n
            if aa > 90:
                while aa > 90:
                    aa -= 26
            word_list_code.append(aa)
        else :
            bb = code + n
            if bb > 122:
                while bb > 122 :
                    bb -= 26
            word_list_code.append(bb)
    for c in word_list_code:
        word_list_chr.append(chr(c))
    return ''.join(word list chr)
```

문제 9번

```python
# 함수 내부에 불필요한 print문이 있는 경우 오답으로 처리가 됩니다.
# 반드시 재귀함수로 구현해야 합니다.
def dec_to_bin(number):
    bin_list = []
    if number >= 2:
        bin_list.append(number % 2)
        dec_to_bin(number//2)
    else:
        if number == 1:
            bin_list.append(1)
        else:
            bin_list.append(0)
    print(bin_list)

# 아래의 코드를 수정하거나 새롭게 추가하지 않습니다.
########## 코드 변경 금지 ############
if __name__ == '__main__':
    print(dec_to_bin(10))  # 1010
    print(dec_to_bin(5))   # 101
    print(dec_to_bin(50))  # 110010
```

```
[1]
[0]
[1]
[0]
None
[1]
[0]
[1]
None
[1]
[1]
[0]
[0]
[1]
[0]
```

문제 9번

```python
def dec_to_bin(number):
    bin_list = []
    def bin(number):
        if number >= 2:
            bin_list.append(f'{number % 2}')
            bin(number//2)
        else:
            if number == 1:
                bin_list.append('1')
            else:
                bin_list.append('0')
    bin(number)
    return ''.join(bin_list[::-1])

# 아래의 코드를 수정하거나 새롭게 추가하지 않습니다.
########## 코드 변경 금지 ############
if __name__ == '__main__':
    print(dec_to_bin(10))  # 1010
    print(dec_to_bin(5))   # 101
    print(dec_to_bin(50))  # 110010
```

```
1010
101
110010
```

문제 9번

```python
def dec_to_bin(number):
    if number == 1:
        return '1'
    elif (number % 2 == 0):
        return dec_to_bin(number//2) + '0'
    elif (number % 2 == 1):
        return dec_to_bin(number//2) + '1'
    else:
        return 0
```

https://pythontutor.com/python-debugger.html#mode=edit

문제 10번

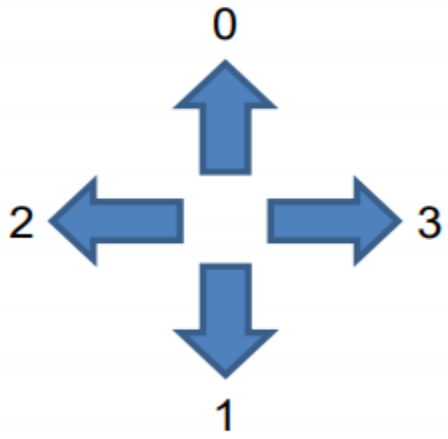
```python
def is_position_safe(N, M, position):
    position_list = [0, 0]
    position_list[0] = position[0]
    position_list[1] = position[1]
    if M == 3:
        position_list[1] += 1
    elif M == 1:
        position_list[0] += 1
    elif M == 2:
        position_list[1] -= 1
    elif M == 0:
        position_list[0] -= 1
    if position_list[0] < 0 or position_list[1] < 0:
        return False
    elif position_list[0] > N or position_list[1] > N:
        return False
    else:
        return True


# 아래의 코드를 수정하거나 새롭게 추가하지 않습니다.
########## 코드 변경 금지 ############
if __name__ == '__main__':
    print(is_position_safe(3, 1, (0, 0)))  # True
    print(is_position_safe(3, 0, (0, 0)))  # False
```

문제 10번

게임 캐릭터가 움직일 수 있는 범위가 있으며, 이 제한된 구역을 넘어가지 않도록 검사하는 함수를 만들려고 한다. 캐릭터는 2차원 평면 (N * N)에서 이동하며, 사용자의 키 입력에 따라 위, 아래, 왼쪽, 오른쪽으로 한 칸씩 움직일 수 있다.
2차원 평면은 list의 내부 요소가 list인 형태를 의미한다.
캐릭터의 현재 위치는 튜플(x, y) 형태로 주어지며, x와 y는 각각 2차원 평면에서의 행과 열을 의미한다. (0 <= x < 100, 0 <= y < 100)
최대 범위는 숫자 N으로 주어진다. (0 < N <= 100)
키 입력은 0부터 3까지의 숫자 M으로 주어지며, 각각 위, 아래, 왼쪽, 오른쪽 방향으로 한 칸 이동을 의미한다.
만약, 키 입력의 결과로 2차원 평면 범위를 벗어난다면 False, 그렇지 않으면 True를 반환하는 함수 is_position_safe를 완성하시오. (반환되는 값 True와 False는 bool 자료형이다.)

(0, 0)	(0, 1)	(0, 2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

좌표 예시



방향 예시

- 입력 예시 : N=3, M=1, position=(0, 0)
- 결과 예시 : True