1. 구간합 문제

```
T = int(input())

for test_case in range(1, T + 1):
    N, M = map(int, input().split())
    arr = list(map(int, input().split()))

    min_sum = sum(arr[:M])
    max_sum = 0

    for i in range(N - M + 1):
        current_sum = sum(arr[i:i + M])

        if current_sum < min_sum:
            min_sum = current_sum

        if current_sum > max_sum:
            max_sum = current_sum

    result = max_sum - min_sum

    print(f'#{test_case} {max_sum - min_sum}')
```

---

2. 숫자를 정렬하자

```
# 버블정렬
T= int(input())

for test_case in range(1, T+1):
    N = int(input())
    arr = list(map(int, input().split()))

    for i in range(N):
        for j in range(i+1, N):
            if arr[i] > arr[j]:
                arr[i], arr[j] = arr[j], arr[i]


    print(f'#{test_case}', *arr)


# 선택정렬
T = int(input())

for test_case in range(1, T+1):
    N = int(input())
    arr = list(map(int, input().split()))
```

```
    for i in range(N-1):
        min_index = i
        for j in range(i+1, N):
            if arr[j] < arr[min_index]:
                min_index = j

        # 현재 위치(i)와 최소값을 가진 위치(min_index)의 원소 교환
        arr[i], arr[min_index] = arr[min_index], arr[i]

    print(f'#{test_case}', *arr)
```

---

3. 연속한 1의 개수

```
T = int(input())
for test_case in range(1, T+1):
    N = int(input())
    arr = list(map(int, input()))

    counts = 0
    ans = 0

    for i in range(N):
        if arr[i] == 1:
            counts += 1
            if counts > ans:
                ans = counts

        if arr[i] != 1:
            counts = 0
    print(f'#{test_case} {ans}')
```

---

4. 풍선팡

```
di = [0, 1, 0, -1]
dj = [1, 0, -1, 0]

T = int(input())
for test_case in range(1, T+1):
    N, M = map(int, input().split())
    arr = [list(map(int, input().split())) for _ in range(N)]

    max_v = 0
    for i in range(N):
        for j in range(M):
            cnt = arr[i][j]
```

```
            for k in range(4):
                ni = i + di[k]
                nj = j + dj[k]
                if 0 <= ni < N and 0 <= nj < M:
                    cnt += arr[ni][nj]

            if cnt > max_v:
                max_v = cnt
    print(f'#{test_case} {max_v}')
```

---

5. 최대 최소의 간격

```
T = int(input())

for test_case in range(1, T+1):
    N = int(input())
    arr = list(map(int, input().split()))

    min_idx = 0
    max_idx = 0



    for i in range(N):
        if arr[min_idx] > arr[i]:
            min_idx = i

        if arr[max_idx] <= arr[i]:
            max_idx = i



    ans = max_idx - min_idx if max_idx > min_idx else min_idx - max_idx
    print(f'#{test_case} {ans}')
```

---

6. 어디에 단어가 들어갈 수 있을까

```
T = int(input())
for test_case in range(1, T+1):
    N, K = map(int, input().split())
    arr = [list(map(int, input().split())) +[0] for _ in range(N)] + [[0]*(N+1)]
    # 가장자리에 0을 추가하는 방법은
    # .split())) 뒤에 +[0], 대괄호 밖에 + [[0]*(N+1)] 추가한다.
    N += 1 # 0인 행과 열 추가

    # 가로 세로로 연속한 1의 개수가 K인 경우를 찾는다.
    ans = 0
```

```
    for i in range(N):
        count = 0   # i행에서 연속된 1의 개수
        for j in range(N):
            if arr[i][j]:   # arr[i][j]==1이면
                count += 1
            else:   # arr[i][j]==0이면
                if count == K:
                    ans += 1
                count =0

    for j in range(N):
        count = 0   # j열에서 연속된 1의 개수
        for i in range(N):
            if arr[i][j]:   # arr[i][j]==1이면
                count += 1
            else:   # arr[i][j]==0이면
                if count == K:
                    ans += 1
                count =0
    print(f'#{test_case} {ans}')
```

---

7. 삼성시 버스 노선

```
T = int(input())

for test_case in range(1, T+1):
    N = int(input())
    counts = [0] * 5001     # 5000번 정류장까지

    # N개의 노선을 정류장에 표시
    for i in range(N):
        A, B = map(int, input().split())
        for j in range(A, B+1):     # 조건 : 1<= A <= B <= 5000
            counts[j] += 1

    P = int(input())
    busstop = [int(input()) for _ in range(P)]

    print(f'#{test_case}', end = ' ')
    for i in busstop:   # 내가 출력할 버스 정류장 번호
        print(counts[i], end = ' ')
```

---

8. 파리퇴치

```
T = int(input())
```

```
for tc in range(1, T+1):
    N, M = list(map(int, input().split()))
    arr = [list(map(int, input().split())) for _ in range (N)]
    max = 0     # 최댓값을 넣을 max

    # 시작점은 NxN 행렬 안의 (i,j)
    for i in range(N):
        for j in range(N):

            # 현재 위치 i,j를 기준으로 M*M 행렬의 합을 구할것
            sum = 0
            # 현재위치 i,j 기준으로 M*M 행렬 정의
            for m1 in range(M):
                for m2 in range(M):
                    # i,j 기준으로 i 를 0에서 (m-1)까지, j 를 0에서 (-1)까지 좌표바
꿈,
                    ni = i + m1
                    nj = j + m2
                    # N*N행렬을 벗어나지 않는 한에서 바꾼 좌표의 arr 값을 sum 에 넣어
줌
                    if 0<= ni < N and 0<= nj <N:
                        sum+= arr[ni][nj]
            if sum > max:
                max = sum
    print(f'#{tc} {max}')
```

9. 숫자 배열 회전

```
def degree_270(list, N):
    result = []
    for j in range(N-1,-1,-1):
        one_line = []
        for k in range(0,N):
            one_line.append(list[k][j])
        result.append(one_line)
    return result

def degree_180(list, N):
    result = []
    for j in range(N-1,-1,-1):
        one_line = []
        for k in range(N-1,-1,-1):
            one_line.append(list[j][k])
        result.append(one_line)
    return result

def degree_90(list, N):
    result = []
    for j in range(0,N):
        one_line = []
        for k in range(N-1,-1,-1):
```

```
            one_line.append(list[k][j])
        result.append(one_line)
    return result

def change_matrix(list, N):
    result = []
    for j in range(0,N):
        one_line = ''
        for k in range(0,N):
            one_line += str(list[j][k])
        result.append(one_line)
    return result
# 입력 값은 아래의 code부터 받는다.

T = int(input())

for tc in range(1,T+1):
    N = int(input()) # N을 받는다.
    base_matrix = [[0] * 3 for _ in range(N)] # 빈 Matrix를 형성한다.

    matrix_info = [] # Matrix에 대한 정보를 받는 과정
    for size in range(0,N):
        matrix_info.append(list(map(int, input().split())))

    matrix_90 = change_matrix(degree_90(matrix_info, N),N)
    matrix_180 = change_matrix(degree_180(matrix_info, N),N)
    matrix_270 = change_matrix(degree_270(matrix_info, N),N)

    for i in range(N):
        base_matrix[i][0] = matrix_90[i]
        base_matrix[i][1] = matrix_180[i]
        base_matrix[i][2] = matrix_270[i]
    print(f"#{tc}")
    for i in range(N):
        for j in range(3):
            print(base_matrix[i][j], end = ' ')
        print()
```

---

10. 부분집합의 합

```
# 1부터 12까지의 숫자를 원소로 가진 집합 A가 있다.
# N = 원소 개수
# K = 원소의 합
# 집합 A의 부분 집합 중 N개의 원소를 갖고 있고, 원소의 합이 K인 부분집합의 개수를 출력하
는 프로그램을 작성하시오.

T = int(input())
for test_case in range(1, T + 1):
    N, K = map(int, input().split())
    A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

```
    n = len(A)
    count = 0

    for i in range(1 << n):   # i : 각 부분집합을 표현  1<<n: 부분집합의 개수
        num_sum = 0     # 원소의 합
        jip_sum = 0     # 원소의 개수

        for j in range(n):  # 원소의 수만큼 비트를 비교
            if i & (1 << j):  # i의 j번 비트가 1인 경우
                num_sum += A[j]  # A 리스트의 j번 원소 더하기
                jip_sum += 1     # 원소의 개수를 1 추가한다.

        if num_sum == K and jip_sum == N:   #원소의 합, 개수가 N, K랑 일치하면
            count += 1      # count를 1 증가시킨다.

    print(f'#{test_case} {count}')
```

---

11. 색칠하기

```
T = int(input())

for test_case in range(1, T + 1):
    N = int(input())
    nemo = [[0] * 10 for _ in range(10)]
    for _ in range(N):
        arr = list(map(int, input().split()))
        A1, A2, B1, B2, color = arr
        for i in range(A1, B1 + 1):
            for j in range(A2, B2 + 1):
                nemo[i][j] += color

    color_bora = 0

    for r in range(10):
        for c in range(10):
            if nemo[r][c] == 3:
                color_bora += 1

    print(f'#{test_case} {color_bora}')
```

---

12. 숫자카드

```
T = int(input())

for test_case in range(1, T+1):
    N = int(input())
    arr = list(map(int, input()))
```

```
    counts = [0] * 10

    for i in range (0, N):
        counts[arr[i]] += 1

    big_num = 0
    number = 0

    for j in range(10):
        if counts[big_num] <= counts[j]:
            number = counts[j]
            big_num = j

    print(f'#{test_case} {big_num} {number}')
```

---

13. 달팽이 숫자

```
t = int(input())

for test_case in range(1, t+1):
    n = int(input())
    num_list = [[0] * n for _ in range(n)]

    # 오른쪽, 아래, 왼쪽, 위
    dx = [0, 1, 0, -1]
    dy = [1, 0, -1, 0]

    x = 0
    y = 0
    i = 0
    count = 1
    num_list[0][0] = 1

    while count < n * n:
        nx = x + dx[i]      # 이동할 곳 출력
        ny = y + dy[i]
        # 이동할 위치가 0보다 작거나, n보다 크거나 같거나, (index 여서 = 추가) 이동할
위치의 값이 0이 아니면.
        if nx < 0 or nx >= n or ny < 0 or ny >= n or num_list[nx][ny] != 0:
            i = (i + 1) % 4 # 방향을 바꿔서 계산하고
            nx = x + dx[i]
            ny = y + dy[i]

        x = nx      # 그렇지 않으면 아까전에 그걸 그냥 그대로 쓰면 돼
        y = ny
        count += 1
        num_list[nx][ny] = count    #증가시킨 다음에 그 증가시킨 값을 여기다가 적어

    print(f'#{test_case}')
    for index in range(n):
```

```
        print(*num_list[index])   # list 없이 값만 출력해줄 때 * 사용 key, value 뽑고 싶을 때는 **
```

---

14. 특별한 정렬

```
T = int(input())

for tc in range(1, T+1):
    N= int(input())
    arr = list(map(int, input().split()))

    for i in range(N):
        if i % 2 == 0:
            max_index = i
            for j in range(i+1,N):
                if arr[j] >= arr[max_index]:
                    max_index = j
            arr[i], arr[max_index] = arr[max_index], arr[i]

        else:
            min_index = i
            for j in range(i+1,N):
                if arr[j] <= arr[min_index]:
                    min_index = j
            arr[i], arr[min_index] = arr[min_index], arr[i]

    print(f'#{tc}', end= ' ')
    for i in range(10):
        print(arr[i], end = ' ')
    print()
```