

섹션2(탐색) [20191222-20191229]

* 탐색의 핵심: 크기를 일일이 비교하여 탐색해야 함

* start, end, idx, mid, target 값을 이용한다

1. 회문 문자열 검사

N개의 문자열 데이터를 입력받아 앞에서 읽을 때나 뒤에서 읽을 때나 같은 경우(회문 문자열)이면 YES를 출력하고 회문 문자열이 아니면 NO를 출력하는 프로그램을 작성한다. 단 회문을 검사할 때 대소문자를 구분하지 않습니다.

▣ 입력설명

첫 줄에 정수 $N(1 \leq N \leq 20)$ 이 주어지고, 그 다음 줄부터 N개의 단어가 입력된다. 각 단어의 길이는 100을 넘지 않는다.

▣ 출력설명

각 줄에 해당 문자열의 결과를 YES 또는 NO로 출력한다.

▣ 입력예제 1

5

level

moon

abcba

soon

gooG

▣ 출력예제 1

#1 YES

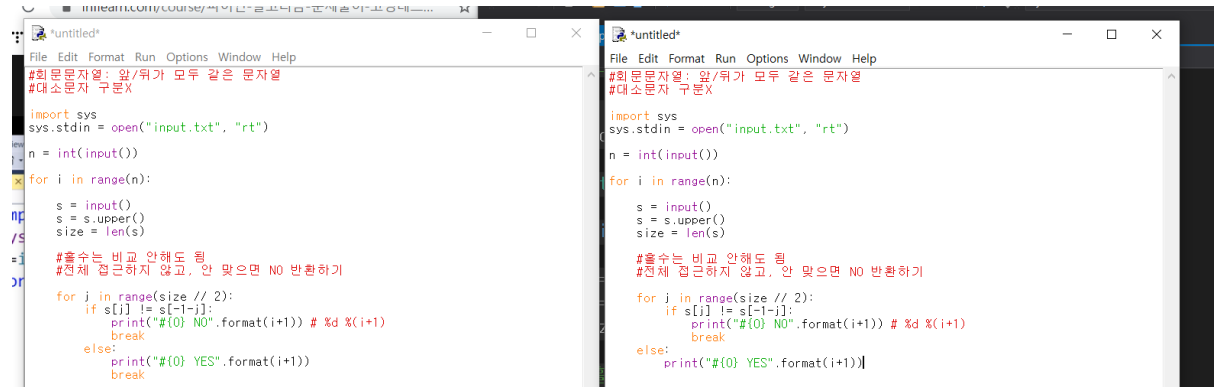
#2 NO

#3 YES

#4 NO

#5 YES

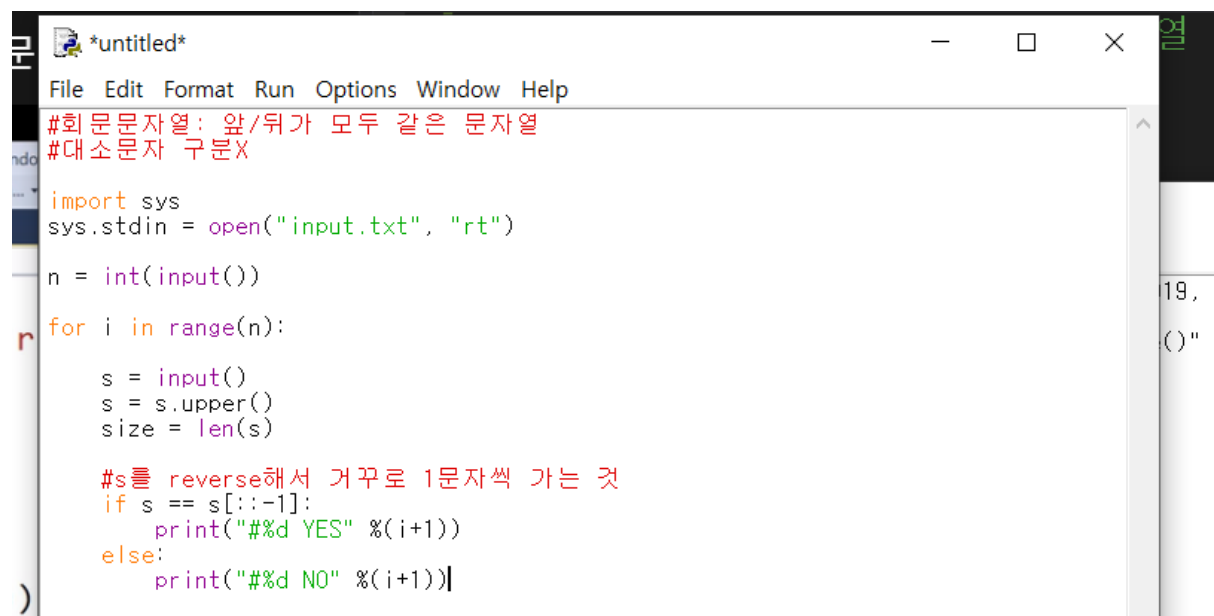
1) size // 2 를 이용한 풀이(면접가서 풀이)



-대소문자 구분을 안 한다고 했으니, 모두 대문자로 바꿔주거나 소문자로 바꿔준 후 문자열 비교

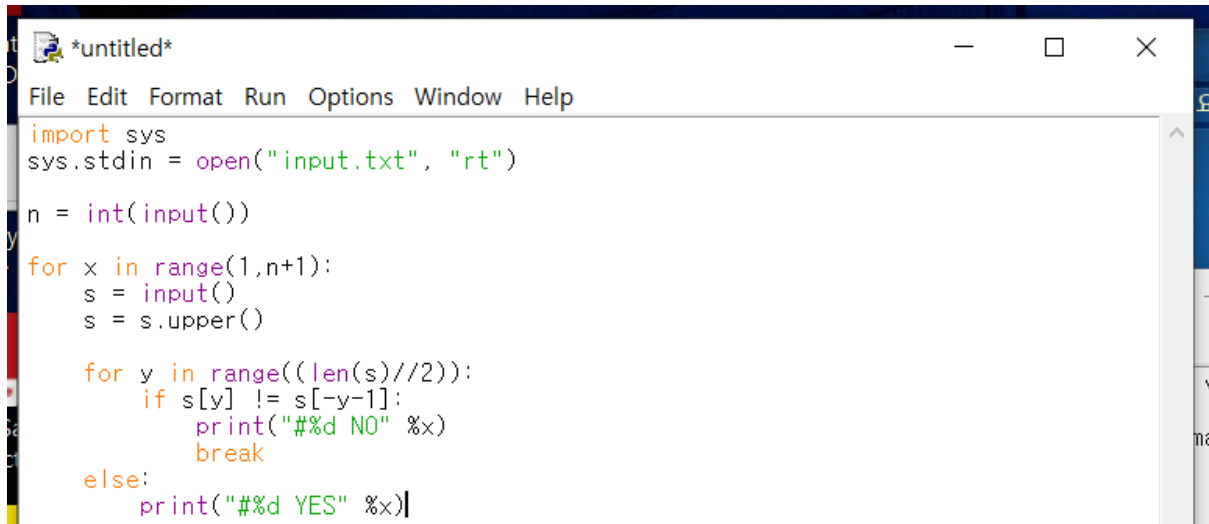
-reverse하여 문자열을 다시 만들 필요 없이, size//2만큼만 비교하여 YES or NO 판단

2) 파이썬스러운 풀이



1) s와 s[::-1]의 비교!

3) 내풀이



```
import sys
sys.stdin = open("input.txt", "rt")

n = int(input())

for x in range(1,n+1):
    s = input()
    s = s.upper()

    for y in range((len(s)//2)):
        if s[y] != s[-y-1]:
            print("#%d NO" %x)
            break
    else:
        print("#%d YES" %x)
```

-문자 개수는 int로 받는다. Input()은 str형

-int형은 range안에 써야 한다. Iterable이 아니라서 list, str처럼 in 다음에 바로 올 수 없다.

-s에 문자열을 받고, 대소문자 구분없이 같으면 회문 문자열이라고 하였으므로, upper로 바꿔준다.

-문자열을 문자열 길이만큼 모두 검사할 필요없이, len(s) // 2만큼만 검사하되, [-1-index]값으로 가운데 값 빼고 양 옆에서 가운데로 오는 문자열이 같은지 확인해준다. For~ else~문 사용

2.숫자만 추출

문자와 숫자가 섞여있는 문자열이 주어지면 그 중 숫자만 추출하여 그 순서대로 자연수를 만듭니다. 만들어진 자연수와 그 자연수의 약수 개수를 출력합니다. 만약 "t0e0a1c2h0er"에서 숫자만 추출하면 0,0,1,2,0이고 이것을 자연수를 만들면 120이 됩니다. 즉 첫 자리 0은 자연수화 할 때 무시합니다. 출력은 120를 출력하고, 다음 줄에 120의 약수의 개수를 출력하면 됩니다. 추출하여 만들어진 자연수는 100,000,000을 넘지 않습니다.

■ 입력설명

첫 줄에 숫자가 섞인 문자열이 주어집니다. 문자열의 길이는 50을 넘지 않습니다.

■ 출력설명

첫 줄에 자연수를 출력하고, 두 번째 줄에 약수의 개수를 출력합니다.

■ 입력예제 1

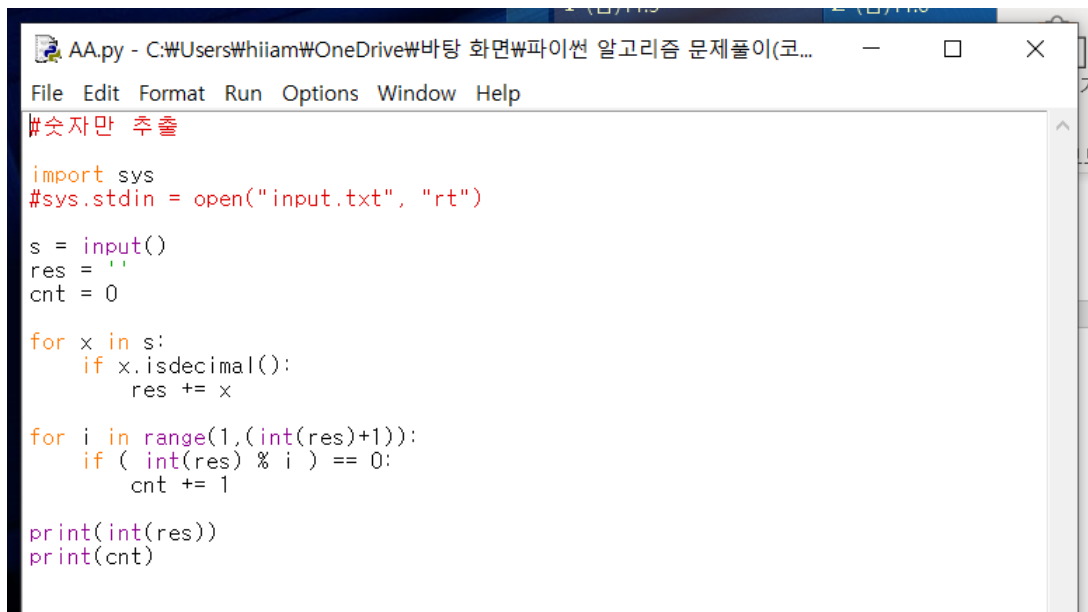
g0en2Ts8eSoft

■ 출력예제 1

28

6

1) 내 풀이



```
AA.py - C:\Users\Whilam\OneDrive\바탕 화면\파이썬 알고리즘 문제풀이(코...
File Edit Format Run Options Window Help
#숫자만 추출

import sys
#sys.stdin = open("input.txt", "rt")

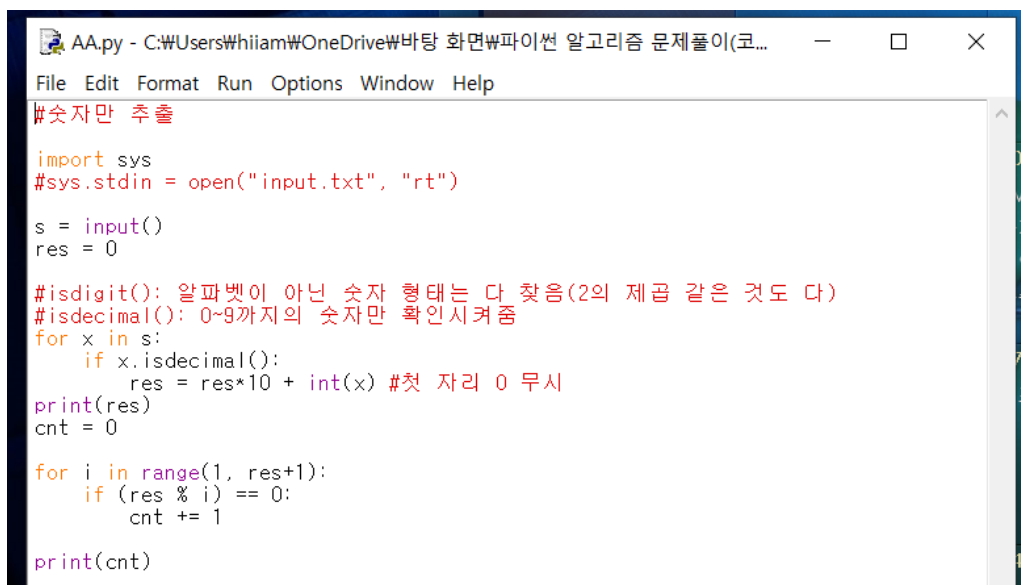
s = input()
res = ''
cnt = 0

for x in s:
    if x.isdecimal():
        res += x

for i in range(1, (int(res)+1)):
    if (int(res) % i) == 0:
        cnt += 1

print(int(res))
print(cnt)
```

2) 선생님 풀이



```
AA.py - C:\Users\Whilam\OneDrive\바탕 화면\파이썬 알고리즘 문제풀이(코...
File Edit Format Run Options Window Help
#숫자만 추출

import sys
#sys.stdin = open("input.txt", "rt")

s = input()
res = 0

#isdigit(): 알파벳이 아닌 숫자 형태는 다 찾음(2의 제곱 같은 것도 다)
#isdecimal(): 0~9까지의 숫자만 확인시켜줌
for x in s:
    if x.isdecimal():
        res = res*10 + int(x) #첫 자리 0 무시
print(res)
cnt = 0

for i in range(1, res+1):
    if (res % i) == 0:
        cnt += 1

print(cnt)
```

-문자열에서 추출한 0을 지우기 위해, $res*10 + int(x)$ 를 사용함

-약수의 개수를 구하기 위해, % 해서 0이 되는 것의 개수를 셈

3. 카드 역배치(정올 기출)

1부터 20까지 숫자가 하나씩 쓰인 20장의 카드가 아래 그림과 같이 오름차순으로 한 줄로 놓여 있다. 각 카드의 위치는 카드 위에 적힌 숫자와 같이 1부터 20까지로 나타낸다. 이제 여러분은 다음과 같은 규칙으로 카드의 위치를 바꾼다: 구간 $[a, b]$ (단, $1 \leq a \leq b \leq 20$)가 주어지면 위치 a 부터 위치 b 까지의 카드를 현재의 역순으로 놓는다.

예를 들어, 현재 카드가 놓인 순서가 위의 그림과 같고 구간이 $[5, 10]$ 으로 주어진다면, 위치 5부터 위치 10까지의 카드 5, 6, 7, 8, 9, 10을 역순으로 하여 10, 9, 8, 7, 6, 5로 놓는다.

이제 전체 카드가 놓인 순서는 아래 그림과 같다. 이 상태에서 구간 $[9, 13]$ 이 다시 주어진다면, 위치 9부터 위치 13까지의 카드 6, 5, 11, 12, 13을 역순으로 하여 13, 12, 11, 5, 6으로 놓는다. 이제 전체 카드가 놓인 순서는 아래 그림과 같다.

오름차순으로 한 줄로 놓여있는 20장의 카드에 대해 10개의 구간이 주어지면, 주어진 구간의 순서대로 위의 규칙에 따라 순서를 뒤집는 작업을 연속해서 처리한 뒤 마지막 카드들의 배치를 구하는 프로그램을 작성하시오.

■ 입력설명

총 10개의 줄에 걸쳐 한 줄에 하나씩 10개의 구간이 주어진다. i 번째 줄에는 i 번째 구간의 시작 위치 a_i 와 끝 위치 b_i 가 차례대로 주어진다. 이때 두 값의 범위는 $1 \leq a_i \leq b_i \leq 20$ 이다.

■ 출력설명

1부터 20까지 오름차순으로 놓인 카드들에 대해, 입력으로 주어진 10개의 구간 순서대로 뒤집는 작업을 했을 때 마지막 카드들의 배치를 한 줄에 출력한다.

■ 입력예제 1

5 10

9 13

1 2

3 4

5 6

1 2

3 4

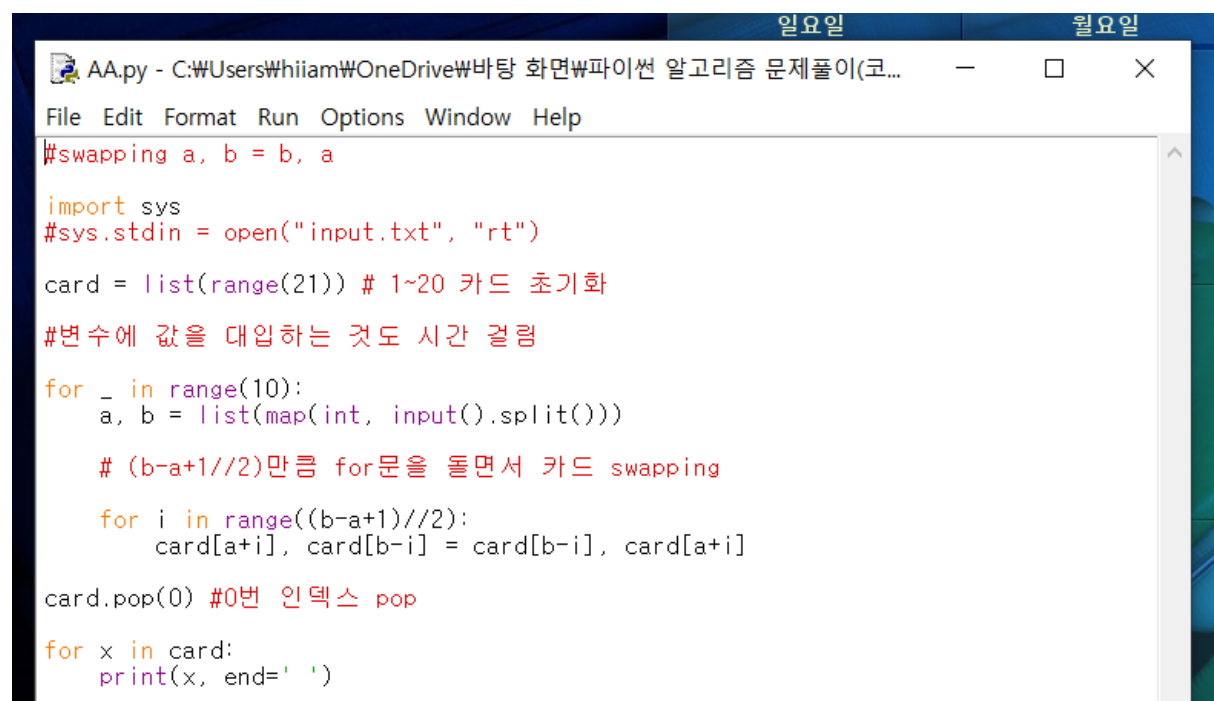
5 6

1 20

1 20

■ 출력예제 1

1 2 3 4 10 9 8 7 13 12 11 5 6 14 15 16 17 18 19 20



```
AA.py - C:\Users\Whiiam\OneDrive\바탕 화면\파이썬 알고리즘 문제풀이(코...
File Edit Format Run Options Window Help
#swapping a, b = b, a

import sys
#sys.stdin = open("input.txt", "rt")

card = list(range(21)) # 1~20 카드 초기화
#변수에 값을 대입하는 것도 시간 걸림

for _ in range(10):
    a, b = list(map(int, input().split()))

    # (b-a+1//2)만큼 for문을 돌면서 카드 swapping

    for i in range((b-a+1)//2):
        card[a+i], card[b-i] = card[b-i], card[a+i]

card.pop(0) #0번 인덱스 pop

for x in card:
    print(x, end=' ')
```

-리스트를 이용해서 0부터 20까지 카드를 만든다

-10번동안 ai, bi를 입력받기 위해 for _ in range(10)을 이용해서 a, b값을 입력받는다.

-그 후, bi랑 ai를 재배치해줘야 하므로 b-a+1//2만큼만 카드를 탐색하면서 서로 swapping해준다.

-0번 인덱스를 pop해서 프린트 해준다.

4. 두 리스트 합치기

오름차순으로 정렬이 된 두 리스트가 주어지면 두 리스트를 오름차순으로 합쳐 출력하는 프로그램을 작성하세요.

■ 입력설명

첫 번째 줄에 첫 번째 리스트의 크기 $N(1 \leq N \leq 100)$ 이 주어집니다.

두 번째 줄에 N 개의 리스트 원소가 오름차순으로 주어집니다.

세 번째 줄에 두 번째 리스트의 크기 $M(1 \leq M \leq 100)$ 이 주어집니다.

네 번째 줄에 M 개의 리스트 원소가 오름차순으로 주어집니다.

각 리스트의 원소는 int형 변수의 크기를 넘지 않습니다.

■ 출력설명

오름차순으로 정렬된 리스트를 출력합니다.

■ 입력예제 1

3

1 3 5

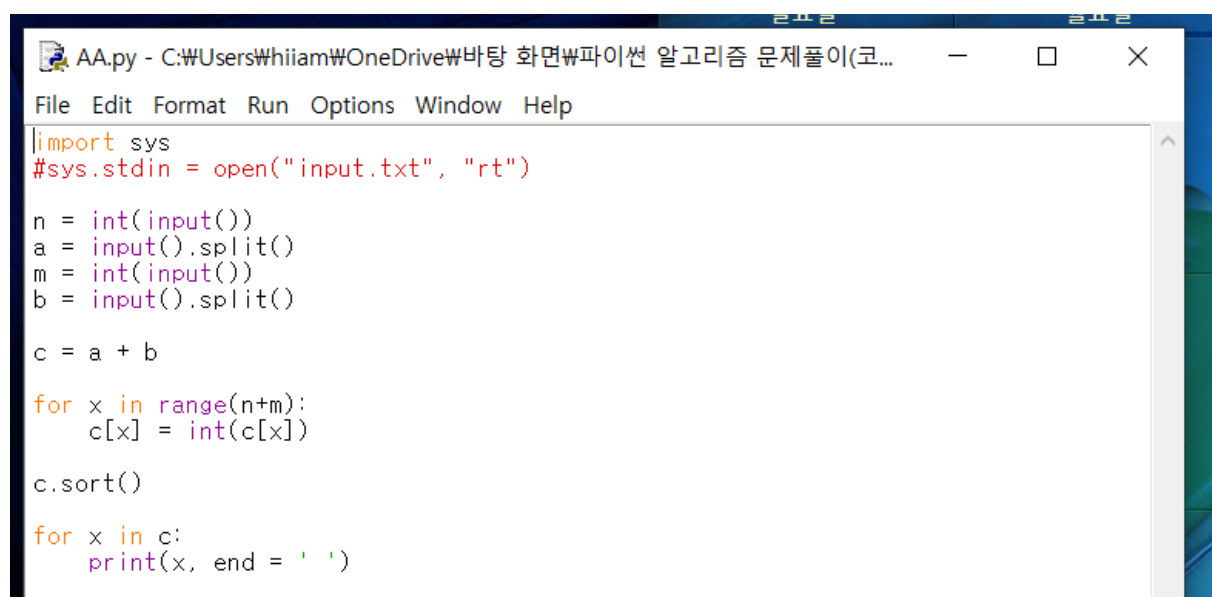
5

2 3 6 7 9

■ 출력예제 1

1 2 3 3 5 6 7 9

1) 내풀이



```
AA.py - C:\Users\Whilam\OneDrive\바탕 화면\파이썬 알고리즘 문제풀이(코...
File Edit Format Run Options Window Help
import sys
#sys.stdin = open("input.txt", "rt")

n = int(input())
a = input().split()
m = int(input())
b = input().split()

c = a + b

for x in range(n+m):
    c[x] = int(c[x])

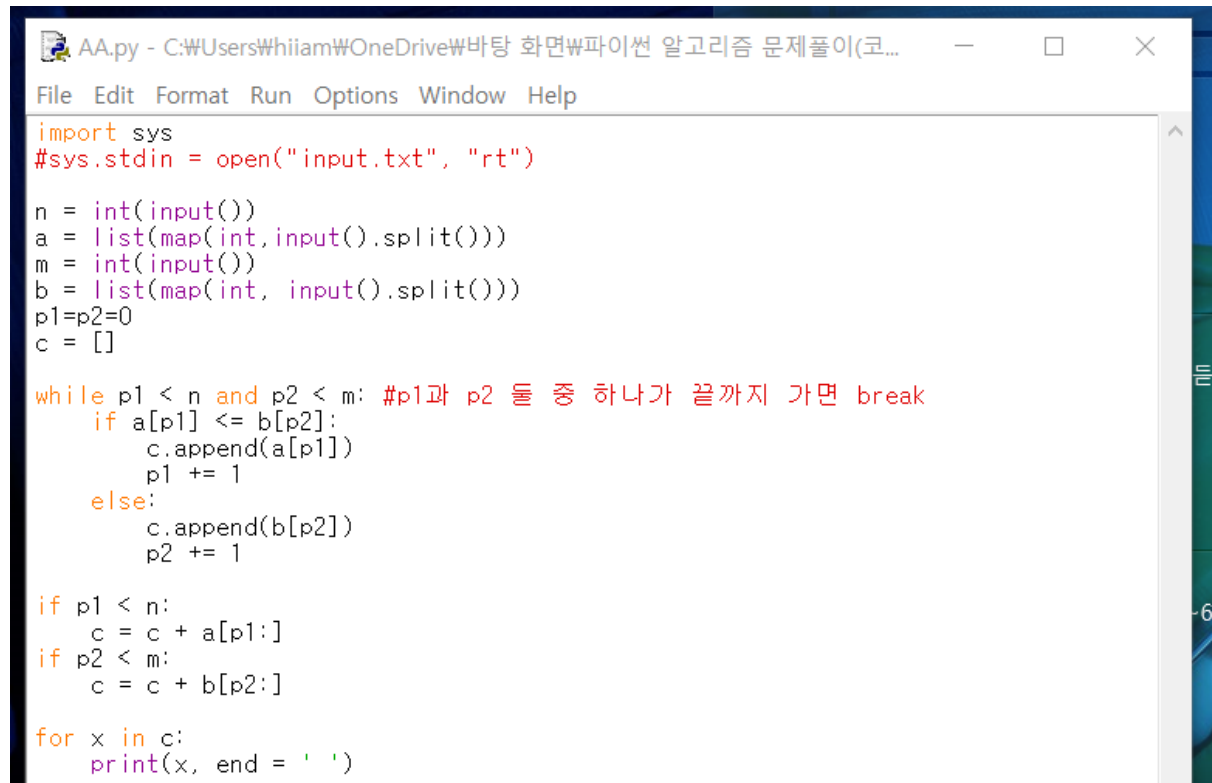
c.sort()

for x in c:
    print(x, end = ' ')
```

2) 선생님 풀이

* `sort()`의 시간 복잡도: 자료가 8개일 때 $8\log 8$, 정렬이 이미 되어있는 것을 정렬로 합치려면 이미 정렬되었다는 사실을 알면 8번만에 자료를 합칠 수 있다. 이 때 $8\log 8$ 과 8에는 큰 차이가 난다.

* n번만에 자료를 합치는 방법



```
import sys
#sys.stdin = open("input.txt", "rt")

n = int(input())
a = list(map(int, input().split()))
m = int(input())
b = list(map(int, input().split()))
p1=p2=0
c = []

while p1 < n and p2 < m: #p1과 p2 둘 중 하나가 끝까지 가면 break
    if a[p1] <= b[p2]:
        c.append(a[p1])
        p1 += 1
    else:
        c.append(b[p2])
        p2 += 1

if p1 < n:
    c = c + a[p1:]
if p2 < m:
    c = c + b[p2:]

for x in c:
    print(x, end = ' ')
```

- 첫 번째 자료를 a리스트에 받고, 두 번째 자료를 b리스트에 받는다.

- 두 자료를 합칠 c리스트를 정의한다.

- a리스트와 b리스트를 인덱싱할 포인터 변수 p1, p2를 0으로 초기화한다.

- a의 p1과 b의 p2를 비교해 나가는데 a의 p1이 가리키는 값이 b의 p2가 가리키는 값보다 작거나 같으면 c에 그 값을 append시키고, 포인터 변수인 p1의 값을 증가시킨다. p1가 p2보다 크면 p2에 있는 값을 append시킴.

- 자료를 다 처리해서 n이나 m까지 갔을 때에는 break시키고 if문을 사용해서 자료를 다 사용한 리스트 확인하고 다 사용하지 않은 리스트는 그냥 c에 붙여버린다.

=>for문으로 하려 했는데 while문을 사용하면 동시에 비교해 나갈 수 있음

5.수들의 합

N 개의 수로 된 수열 $A[1], A[2], \dots, A[N]$ 이 있다. 이 수열의 i 번째 수부터 j 번째 수까지의 합 $A[i]+A[i+1]+\dots+A[j-1]+A[j]$ 가 M 이 되는 경우의 수를 구하는 프로그램을 작성하시오.

▣ 입력설명

첫째 줄에 $N(1 \leq N \leq 10,000)$, $M(1 \leq M \leq 300,000,000)$ 이 주어진다. 다음 줄에는 $A[1], A[2], \dots, A[N]$ 이 공백으로 분리되어 주어진다. 각각의 $A[x]$ 는 30,000을 넘지 않는 자연수이다.

▣ 출력설명

첫째 줄에 경우의 수를 출력한다.

▣ 입력예제 1

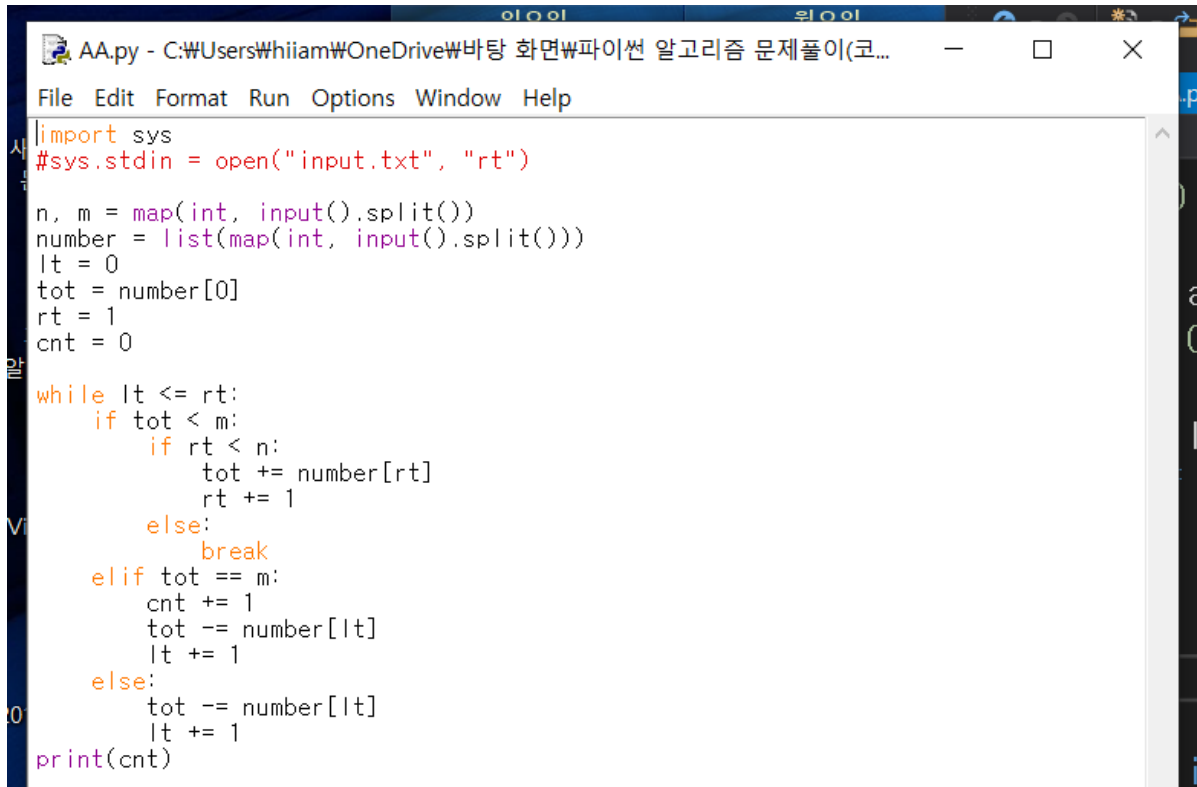
8 3

1 2 1 3 1 1 1 2

▣ 출력예제 1

5

* $i \sim j$ 까지 연속된 구간이 M 이 되는 경우의 수를 구하는 문제



```
AA.py - C:\Users\Whiiam\OneDrive\바탕 화면\파이썬 알고리즘 문제풀이(코...
File Edit Format Run Options Window Help

import sys
#sys.stdin = open("input.txt", "rt")

n, m = map(int, input().split())
number = list(map(int, input().split()))
lt = 0
tot = number[0]
rt = 1
cnt = 0

while lt <= rt:
    if tot < m:
        if rt < n:
            tot += number[rt]
            rt += 1
        else:
            break
    elif tot == m:
        cnt += 1
        tot -= number[lt]
        lt += 1
    else:
        tot -= number[lt]
        lt += 1

print(cnt)
```

- 숫자배열을 list인 a로 만들고, 시작점 lt와 끝 점 rt, 숫자를 더할 tot, 카운트 변수 cnt를 선언
- lt를 a의 첫 번째 원소로 정하고, rt를 한 칸씩 늘려가며 cnt를 세는데 lt가 rt를 넘어가면 break
- tot가 목표하는 target 숫자인 m이 안 되었을 때, rt가 n이 안되었다면 tot에 rt값을 계속 더해주고, n이 되었는데 m이 안 되었다면 break!
- tot가 목표하는 target 숫자인 m이 되었다면, cnt 에 1을 더해주고, lt는 그 다음 값이 된다.
- tot가 목표하는 target 숫자를 넘었다면 그냥 lt는 그 다음 값이 된다.

6. 격자판 최대합

5*5 격자판에 아래와 같이 숫자가 적혀있습니다. N*N의 격자판이 주어지면 각 행의 합, 각 열의 합, 두 대각선의 합 중 가 장 큰 합을 출력합니다.

■ 입력설명

첫 줄에 자연수 N이 주어진다.(1<=N<=50)

두 번째 줄부터 N줄에 걸쳐 각 줄에 N개의 자연수가 주어진다. 각 자연수는 100을 넘지 않는다.

■ 출력설명

최대합을 출력합니다.

■ 입력예제 1

5

10 13 10 12 15

12 39 30 23 11

11 25 50 53 15

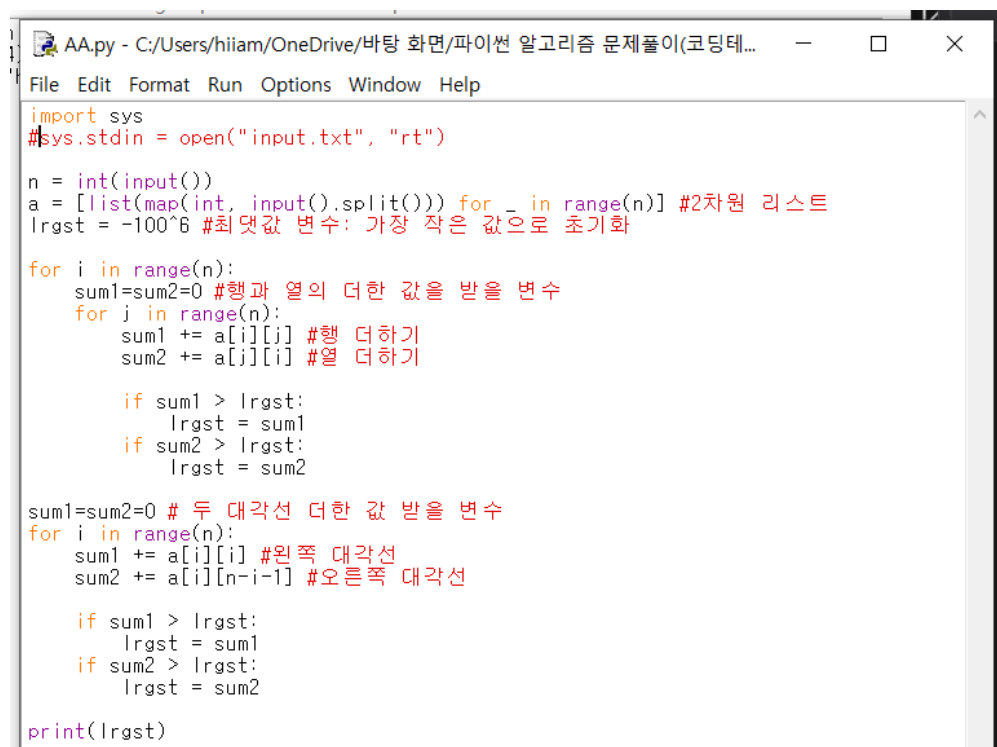
19 27 29 37 27

19 13 30 13 19

■ 출력예제 1

155

1) 선생님풀이



```
import sys
#sys.stdin = open("input.txt", "rt")

n = int(input())
a = [list(map(int, input().split())) for _ in range(n)] #2차원 리스트
lrgst = -100^6 #최댓값 변수: 가장 작은 값으로 초기화

for i in range(n):
    sum1=sum2=0 #행과 열의 더한 값을 받을 변수
    for j in range(n):
        sum1 += a[i][j] #행 더하기
        sum2 += a[j][i] #열 더하기

        if sum1 > lrgst:
            lrgst = sum1
        if sum2 > lrgst:
            lrgst = sum2

sum1=sum2=0 # 두 대각선 더한 값 받을 변수
for i in range(n):
    sum1 += a[i][i] #왼쪽 대각선
    sum2 += a[i][n-i-1] #오른쪽 대각선

    if sum1 > lrgst:
        lrgst = sum1
    if sum2 > lrgst:
        lrgst = sum2

print(lrgst)
```

-2차원 리스트로 격자판들을 입력받는다.

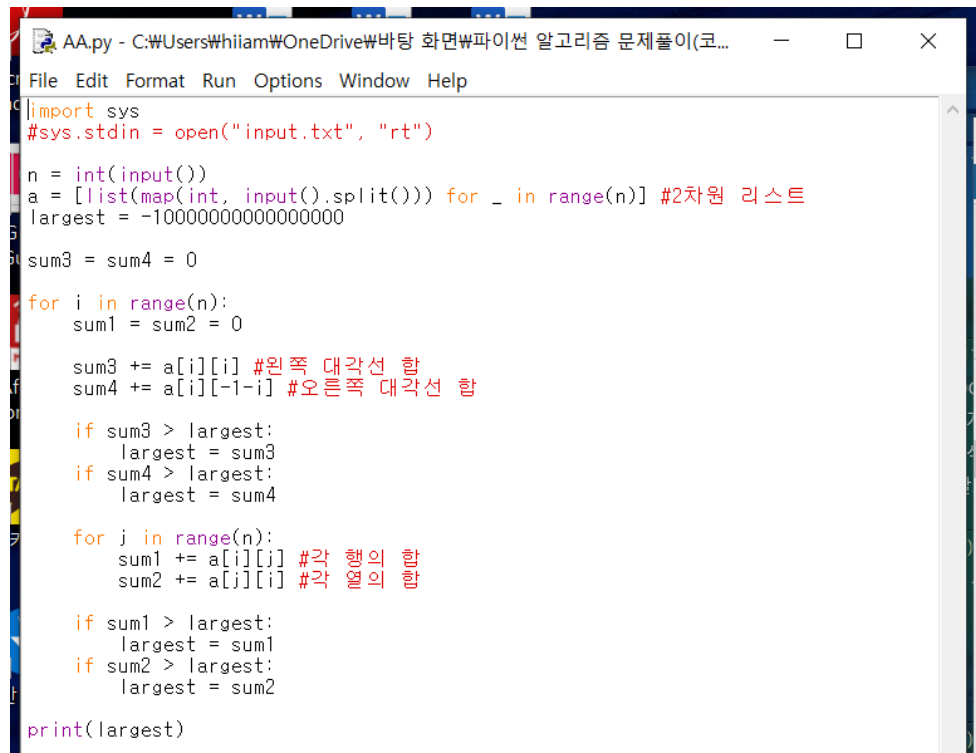
-최댓값 변수를 받을 lrgst를 가장 작은 값으로 초기화해준다.

-2중 for문을 돌면서, 각 행을 더한 값을 sum1에 저장하고 각 열을 더한 값을 sum2에 저장해서

최댓값이면 lrgst에 저장해준다.

- for문을 돌면서 왼쪽 대각선을 더한 값을 sum1에 저장하고 오른쪽 대각선을 더한 값을 sum2에 저장해서 최댓값이면 lrgst에 저장해준다.

2) 내 풀이



```
import sys
#sys.stdin = open("input.txt", "rt")

n = int(input())
a = [list(map(int, input().split())) for _ in range(n)] #2차원 리스트
largest = -1000000000000000000

sum3 = sum4 = 0

for i in range(n):
    sum1 = sum2 = 0

    sum3 += a[i][i] #왼쪽 대각선 합
    sum4 += a[i][-1-i] #오른쪽 대각선 합

    if sum3 > largest:
        largest = sum3
    if sum4 > largest:
        largest = sum4

    for j in range(n):
        sum1 += a[i][j] #각 행의 합
        sum2 += a[j][i] #각 열의 합

    if sum1 > largest:
        largest = sum1
    if sum2 > largest:
        largest = sum2

print(largest)
```

-sum3, sum4를 for문 안에 선언해버리면 대각선 합들이 다시 0이 된다. 그래서 밖에 써주어야 하고, [n-i-1]이 아니라 [-i-1]이라고 하면 편하다.

7. 사과나무(다이아몬드)

현수의 농장은 $N \times N$ 격자판으로 이루어져 있으며, 각 격자안에는 한 그루의 사과나무가 심어져있다. N 의 크기는 항상 홀수이다. 가을이 되어 사과를 수확해야 하는데 현수는 격자판안의 사과를 수확할 때 다이아몬드 모양의 격자판만 수확하고 나머지 격자안의 사과는 새들을 위해서 남겨놓는다. 만약 N 이 5이면 아래 그림과 같이 진한 부분의 사과를 수확한다.

현수와 수확하는 사과의 총 개수를 출력하세요.

■ 입력설명

첫 줄에 자연수 N (홀수)이 주어진다.($3 \leq N \leq 20$)

두 번째 줄부터 N줄에 걸쳐 각 줄에 N개의 자연수가 주어진다.

이 자연수는 각 격자안에 있는 사과나무에 열린 사과의 개수이다.

각 격자안의 사과의 개수는 100을 넘지 않는다.

■ 출력설명

수확한 사과의 총 개수를 출력합니다.

■ 입력예제 1

5

10 13 10 12 15

12 39 30 23 11

11 25 50 53 15

19 27 29 37 27

19 13 30 13 19

■ 출력예제 1

379

1) 내풀이

```
AA.py - C:\Users\Whiam\OneDrive\바탕 화면\파이썬 알고리즘 문제풀이(코...
File Edit Format Run Options Window Help

import sys
#sys.stdin = open("input.txt", "rt")

n = int(input())
a = [list(map(int, input().split())) for _ in range(n)] # 격자판

cnt = 0 #수확하는 사과의 갯수
idx1 = 0
idx2 = 1

for i in range(n):
    if i < ( n // 2 ):
        for j in (a[i][(n//2)-idx1:(n//2)+idx2]):
            cnt += j
            idx1 += 1
            idx2 += 1

    elif i >= ( n // 2 ):
        for j in (a[i][(n//2)-idx1:(n//2)+idx2]):
            cnt += j
            idx1 -= 1
            idx2 -= 1

print(cnt)
```

2) 선생님 풀이

```
*untitled*
File Edit Format Run Options Window Help

import sys
#sys.stdin = open("input.txt", "rt")

n = int(input())
a = [list(map(int, input().split())) for _ in range(n)] # 격자판

res = 0 #수확하는 사과 갯수
s = e = n//2 #처음과 끝 지정

for i in range(n):
    for j in range(s, e+1): #s부터 e까지 다이아몬드 탐색
        res += a[i][j]

    if i < n//2:
        s -= 1
        e += 1
    else:
        s += 1
        e -= 1

print(res)
```

-수확하는 사과의 개수를 res=0으로 초기화

-start와 end를 처음에 n//2로 초기화

-행을 옮길 때 마다 start node는 1을 감소시키고, end node는 1을 증가시킴

-처음에 start와 end값을 for문을 돌기 전에 설정해주면 좀 더 간편하다

8. 곳감(모래시계)

현수는 곳감을 만들기 위해 감을 깎아 마당에 말리고 있습니다. 현수의 마당은 $N \times N$ 격자판으로 이루어져 있으며, 현수는 각 격자단위로 말리는 감의 수를 정합니다. 그런데 해의 위치에 따라 특정위치의 감은 잘 마르지 않습니다. 그래서 현수는 격자의 행을 기준으로 왼쪽, 또는 오른쪽으로 회전시켜 위치를 변경해 모든 감이 잘 마르게 합니다. 만약 회전명령 정보가 2 0 3이면 2번째 행을 왼쪽으로 3만큼 아래 그림처럼 회전시키는 명령입니다.

첫 번째 수는 행번호, 두 번째 수는 방향인데 0이면 왼쪽, 1이면 오른쪽이고, 세 번째 수는 회전하는 격자의 수입니다. M개의 회전명령을 실행하고 난 후 아래와 같이 마당의 모래시계 모양의 영역에는 감이 총 몇개가 있는지 출력하는 프로그램을 작성하세요.

■ 입력설명

첫 줄에 자연수 $N(3 \leq N \leq 20)$ 이 주어져, N은 홀수입니다.

두 번째 줄부터 N줄에 걸쳐 각 줄에 N개의 자연수가 주어진다.

이 자연수는 각 격자안에 있는 감의 개수이며, 각 격자안의 감의 개수는 100을 넘지 않는다.

그 다음 줄에 회전명령의 개수인 $M(1 \leq M \leq 10)$ 이 주어지고, 그 다음 줄부터 M개의 회전명령 정보가 M줄에 걸쳐 주어집니다.

- 10 -

■ 출력설명

총 감의 개수를 출력합니다.

■ 입력예제 1

5

10 13 10 12 15

12 39 30 23 11

11 25 50 53 15

19 27 29 37 27

19 13 30 13 19

3

2 0 3

5 1 2

3 1 4

▣ 출력예제 1

362

1) 내폴이


```
AA.py - C:\Users\Whilam\OneDrive\바탕 화면\파이썬 알고리즘 문제풀이(코...
File Edit Format Run Options Window Help

import sys
from collections import deque
#sys.stdin = open("input.txt", "rt")

n = int(input())
a = [list(map(int, input().split())) for _ in range(n)] #격자판 리스트
m = int(input()) #회전명령 갯수

for i in range(m):
    x, y, z = map(int, input().split()) #행, 방향, 칸

    for j in range(n):

        if j == x-1: #회전명령 행이면
            dq = deque(a[j])

            if y == 0: #왼쪽 방향
                for k in range(z): #z칸동안 회전
                    cur = dq.popleft()
                    dq.append(cur)
                a[j] = list(dq)

            elif y == 1: #오른쪽 방향
                for k in range(z): #z칸동안 회전
                    cur = dq.pop()
                    dq.appendleft(cur)
                a[j] = list(dq)

res = 0
s = 0
e = n
for i in range(n):
    for j in range(s,e):
        res += a[i][j]

    if i < ( n // 2 ):
        s += 1
        e -= 1

    else:
        s -= 1
        e += 1

print(res)
```

-queue를 이용해서 회전명령을 시켰으나, list 자체에서 pop()과 pop(0)과 insert(idx, 요소), append()를 이용할 수 있음

-n//2 미만이어야 하는데, 자꾸 이하라고 해서 오답이 나왔음 잘 체크할 것

2) 선생님 풀이

```
AA.py - C:\Users\Whilam\OneDrive\바탕 화면\파이썬 알고리즘 문제풀이(코...
File Edit Format Run Options Window Help

import sys
#sys.stdin = open("input.txt", "rt")

n = int(input())
a = [list(map(int, input().split())) for _ in range(n)] #격자판 리스트
m = int(input()) #회전명령 갯수

for i in range(m):
    h, t, k = map(int, input().split()) #행, 방향, 칸

    if t == 0: #왼쪽
        for _ in range(k): #k칸만큼 회전
            a[h-1].append(a[h-1].pop(0)) #제일 앞 데이터를 맨 뒤에 넣음
    elif t == 1: #오른쪽
        for _ in range(k):
            a[h-1].insert(0, a[h-1].pop()) #제일 뒤 데이터를 맨 앞에 넣음

res=0
s=0
e=n-1

for i in range(n):
    for j in range(s, e+1):
        res += a[i][j]

    if i < n//2:
        s += 1
        e -= 1

    else:
        s -= 1
        e += 1

print(res)
```

9. 봉우리

지도 정보가 $N \times N$ 격자판에 주어집니다. 각 격자에는 그 지역의 높이가 쓰여있습니다. 각 격자판의 숫자 중 자신의 상하좌우 숫자보다 큰 숫자는 봉우리 지역입니다. 봉우리 지역이 몇 개 있는지 알아내는 프로그램을 작성하세요. 격자의 가장자리는 0으로 초기화 되었다고 가정한다. 만약 $N=5$ 이고, 격자판의 숫자가 다음과 같다면 봉우리의 개수는 10개입니다.

■ 입력설명

첫 줄에 자연수 N 이 주어진다. ($1 \leq N \leq 50$)

두 번째 줄부터 N 줄에 걸쳐 각 줄에 N 개의 자연수가 주어진다. 각 자연수는 100을 넘지 않는다.

■ 출력설명

봉우리의 개수를 출력하세요.

▣ 입력예제 1

5

5 3 7 2 3

3 7 1 6 1

7 2 5 3 4

4 3 6 4 1

8 7 3 5 2

▣ 출력예제 1

10