(1) 칵테일 쉐이커 정렬 알고리즘

--------------------------------------------------------------------

cocktailShakerSort(a[], n)

d ← true; i ← 1; k ← n;

while (i ≤ k) do {

if (d = true) then {

for (j ← i; j < k; j ← j+1) do {

if (a[j] > a[j+1]) then a[j]와 a[j+1] 교환;

k ← k - 1;

}

}

else {

for (j ← k; j > i; j ← j-1) do {

if (a[j] < a[j-1]) then a[j]와 a[j-1] 교환;

i ← i + 1;

}

}

d ← not d;

}

end cocktailShakerSort()

--------------------------------------------------------------------

□ 파이썬 소스 코드

--------------------------------------------------------------------

def cocktailShaker(a, n):

d = True

i, k = 1, n

while i <= k:

if d:

for j in range(i, k):

if a[j] > a[j+1]:

a[j], a[j+1] = a[j+1], a[j]

k -= 1

else:

for j in range(k, i, -1):

if a[j] < a[j-1]:

a[j], a[j-1] = a[j-1], a[j]

i += 1

d = not d

print(a)

a = [None, 6, 5, 4, 3, 2, 1]

cocktailShaker(a, 6)

--------------------------------------------------------------------

(2) 교환 정렬 알고리즘

--------------------------------------------------------------------

exchangeSort(a[], n)

for (i ← 1; i < n; i ← i+1) do

for (j ← i+1; j ≤ n; j ← j+1) do

if (a[i] < a[j]) then a[i]와 a[j] 교환;

exchangeSort(a[], n)

--------------------------------------------------------------------

□ 파이썬 소스 코드

--------------------------------------------------------------------

def exchangeSort(a, n):

for i in range(1, n):

for j in range(i+1, n+1):

if a[i] < a[j]:

a[i], a[j] = a[j], a[i]

print(a)

a = [None, 3, 1, 2, 4, 6, 5]

exchangeSort(a, 6)

--------------------------------------------------------------------

(2) 자연 합병 정렬 알고리즘

--------------------------------------------------------------------

makeRun(a[], n)

i ← 1;

r ← 공백 리스트;

while (i ≤ n) do {

t ← 공백 리스트;

t에 a[i] 삽입;

while (i+1 < n and a[i] ≤ a[i+1]) do {

t에 a[i+1] 삽입;

i ← i + 1;

}

r에 t 삽입

i ← i + 1;

return r;

end makeRun()

--------------------------------------------------------------------

□ 파이썬 소스 코드

--------------------------------------------------------------------

def merge(r1, r2):

i, j, n1, n2 = 0, 0, len(r1), len(r2)

b = []

while i < n1 and j < n2:

if r1[i] < r2[j]:

b.append(r1[i])

i += 1

else:

b.append(r2[j])

j += 1

if i == n1:

for k in range(j, n2):

b.append(r2[k])

else:

for k in range(i, n1):

b.append(r1[k])

return b

def makeRun(a, n):

i = 1

r = []

while i <= n:

t = []

t.append(a[i])

while i+1 < n and a[i] <= a[i+1]:

t.append(a[i+1])

i += 1

r.append(t)

i += 1

return r

def naturalMerge(a, n):

r = makeRun(a, n)

print(r)

m = len(r)

while m > 1:

i = 0

p = []

while i < m:

if i == m-1:

p.append(r[i])

else:

p.append(merge(r[i], r[i+1]))

i += 2

r = []

m = len(p)

for j in range(m):

r.append(p[j])

for i in range(n):

a[i+1] = r[0][i]

return a

a = [None, 6, 7, 8, 3, 4, 1, 5, 9, 10, 2]

print(naturalMerge(a, 10))

--------------------------------------------------------------------

**<코딩 테스트 연습 #2> 나머지 합**

□ 파이썬 소스 코드

--------------------------------------------------------------------

N, M = map(int, input().split())

A = list(map(int, input().split()))

S = [0] \* N

S[0] = A[0]

for i in range(1, N):

S[i] = S[i-1] + A[i]

print()

#print('S = %s'%S)

count = 0

C = [0] \* M

for i in range(N):

R = S[i] % M

if R == 0:

count += 1

C[R] += 1

#print('C = %s'%C)

for i in C:

if i > 0:

count += i \* (i - 1) // 2

print(count)

--------------------------------------------------------------------