(1) 에라토스테네스의 체

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

eratos(a, n)

a[1] ← 0;

for (i ← 2; i < n/2; i ← i + 1) do

for (j ← 2; i×j ≤ n; j ← j + 1) do

a[i×j] ← 0;

return a;

end eratos()

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

(2) 두 수의 합

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

twoSum(a[], n, t)

for (i ← 0; i < n; i ← i + 1) do

for (j ← i + 1; j < n; j ← j + 1) do

if (a[i] + a[j] = t) then

print(i+1 번째와 j+1 번째 원소);

end twoSum()

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

(3) 파스칼의 수 삼각형

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

pascalTriangle(h)

res ← 공백 리스트;

pre ← 공백 리스트;

pre에 1 추가;

res에 pre 추가;

for (i ← 1; i < h; i ← i + 1) do {

cur ← 공백 리스트;

cur에 1 추가;

for (j ← 0; j < pre의 길이 - 1; j ← j + 1) do

cur에 pre[j]와 pre[j+1]의 합 추가;

cur에 1 추가;

res에 cur 추가;

pre ← cur;

return res;

end pascalTriangle()

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

(4) 최장 연속 순차

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

longestSequence(a[], s, n)

max\_len ← 1;

for (i ← 0; i < n; i ← i + 1) do {

left ← a[i] - 1;

right ← a[i] + 1;

count ← 1;

while (left ∈ s) do {

count ← count + 1;

s에서 left 제거;

left ← left - 1;

}

while (right ∈ s) do {

count ← count + 1;

s에서 right 제거;

right ← right + 1;

}

max\_len ← max(max\_len, count);

}

return max\_len;

end longestSequence()

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

(5) 이진 탐색 구현

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

def binarySearch(a, key, left, right):

if left <= right:

mid = (left + right) // 2

if key == a[mid]:

return mid

elif key < a[mid]:

return binarySearch(a, key, left, mid-1)

else:

return binarySearch(a, key, mid+1, right)

else:

return -1

import random

A = []

for i in range(10):

A.append(random.randint(1, 50))

A.sort()

print(A)

key = int(input('탐색 키 입력 : '))

res = binarySearch(A, key, 0, len(A) - 1)

if res == -1:

print('탐색 키를 갖는 원소를 찾을 수 없음')

else:

print(res+1, '번째 원소')

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