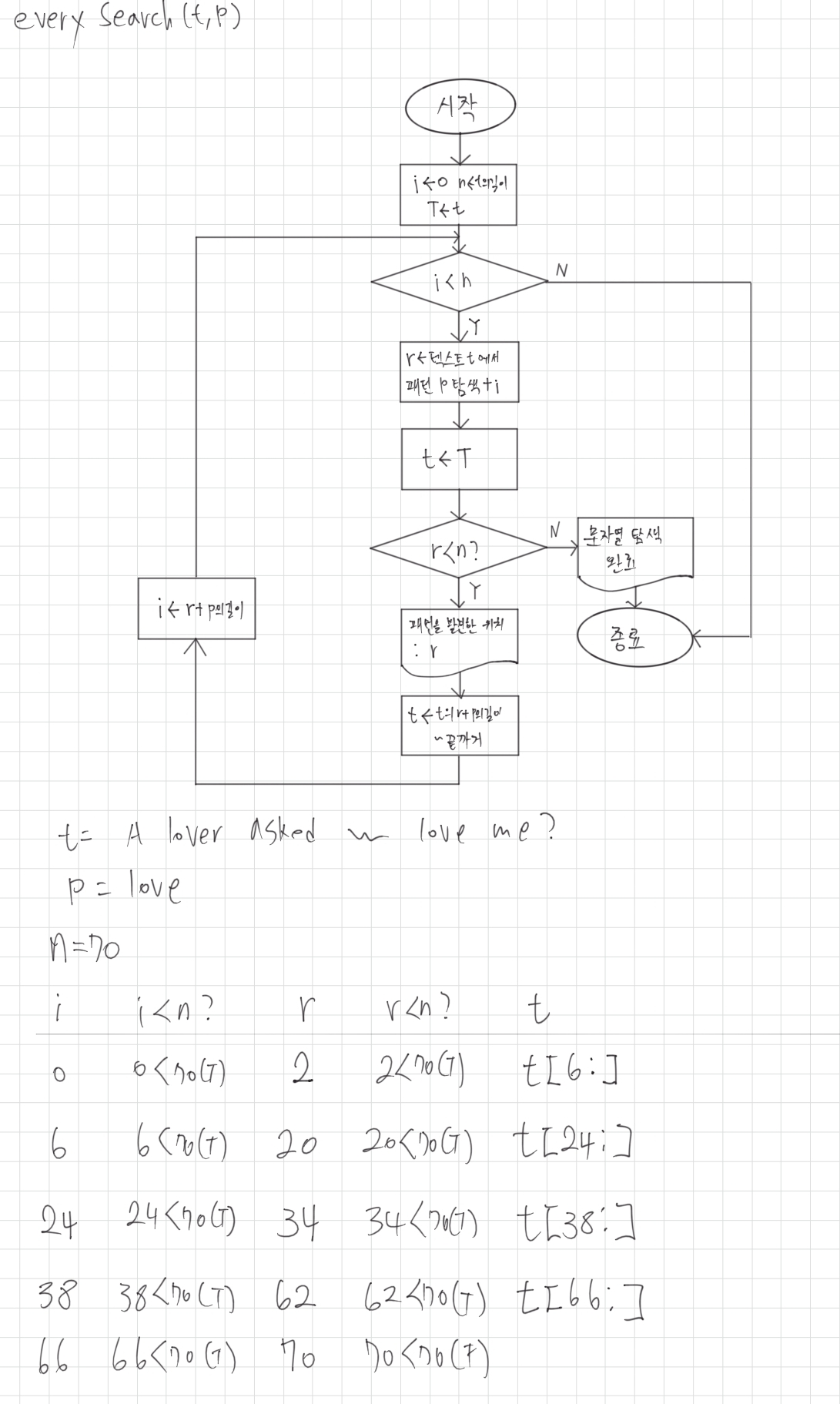
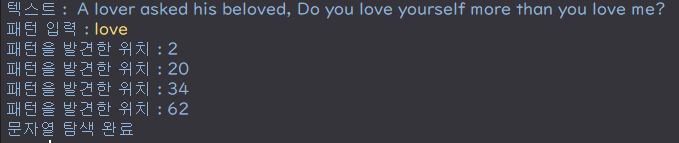
1. 텍스트에 패턴이 여러 개 있을 경우 패턴을 발견한 위치를 모두 찾아서 출력해 주는 함수 everySearch(t, p)에 대한 순서도를 작성하고, 이를 사용하여 파이썬 프로그램을 작성하 라. everySearch(t, p) 함수에서 함수 stringSearch(t, p)를 호출하여 문자열 탐색을 수행 한다. stringSearch(t, p)는 수업 시간에 배운 함수를 그대로 사용하고, 이 함수를 순서도 에서 표현할 때는 “텍스트 t에서 패턴 p 탐색”이라고 하면 된다. stringSearch(t, p)에 대 한 순서도는 그리지 않아도 된다.

<순서도와 표>



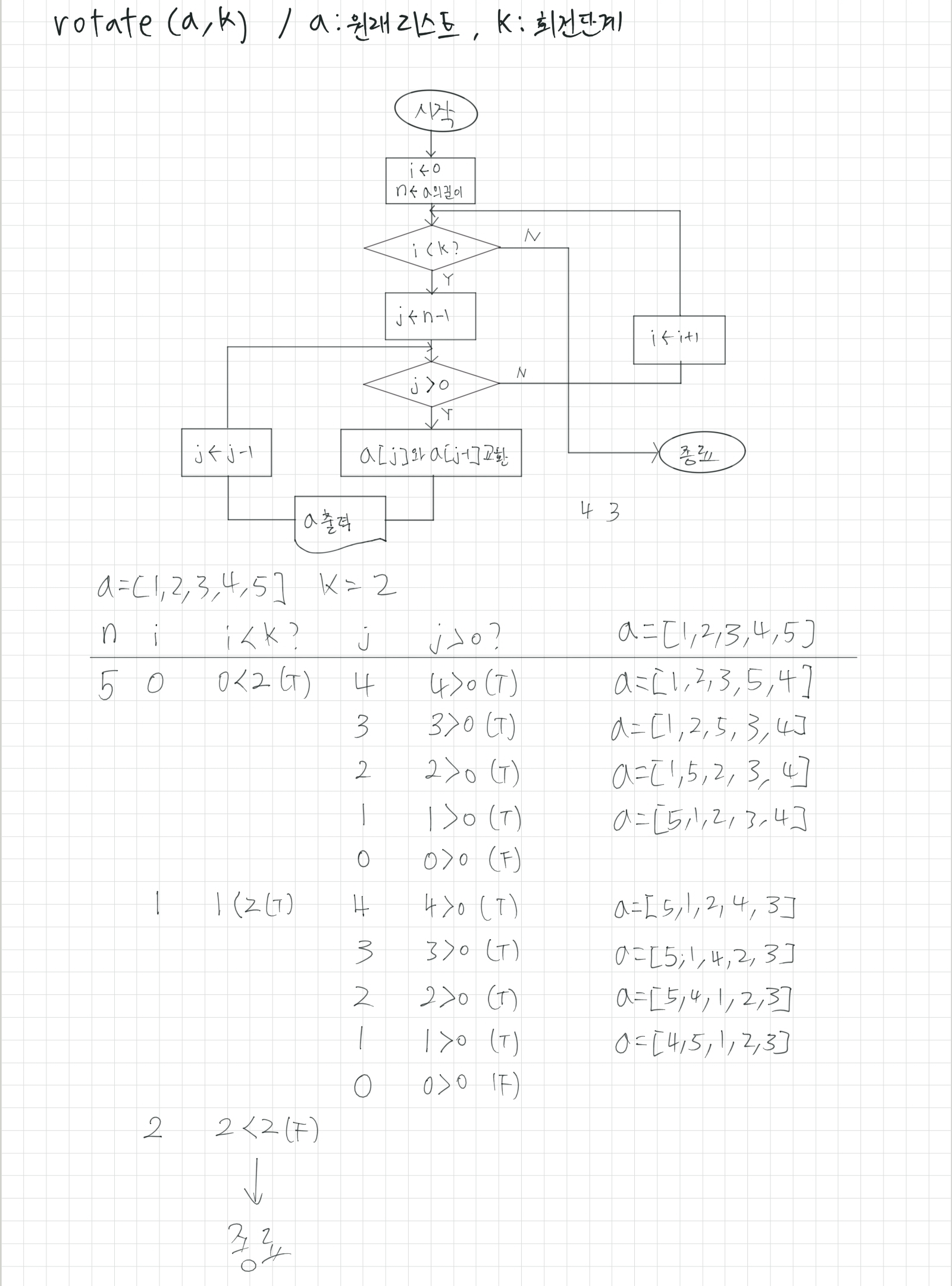
<코드와 실행결과>

|  |
| --- |
| def stringSearch(t, p):  n, m = len(t), len(p)  i, j = 0, 0  while i < n and j < m:  if t[i] != p[j]:  i -= j  j = -1  i += 1  j += 1  if j == m : return i-m  else : return i  def everySearch(t, p):  i = 0  n = len(t)  T = t  while i < n:  r = stringSearch(t, p) + i  t = T  if r < n:  print('패턴을 발견한 위치 : %d'%(r))  else : print('문자열 탐색 완료')  t = t[r+len(p):]  i = r + len(p)  print('텍스트 : ', end='')  f = open('input.txt', 'r')  line = f.readline()  print(line, end='')  p = input('패턴 입력 : ')  everySearch(line, p) |



2. 추가 리스트를 사용하지 않고 원래 리스트 a에서 원소를 k 단계만큼 회전시키는 함수 rotate(a, k)에 대한 순서도를 그리고, 이를 사용하여 파이썬 프로그램을 작성하라.

<순서도와 표>



<코드와 실행결과>

|  |
| --- |
| def rotate (a, k):  n = len(a)  for i in range(k):  for j in range(n-1, 0, -1):  a[j], a[j-1] = a[j-1], a[j]  print(a)  n = int(input('원소의 개수 : '))  k = int(input('회전 단계수 : '))  a = list(range(1, n+1))  print('원래 리스트 : ', a)  rotate(a, k)  print('회전 리스트 : ', a) |

