



PYTHON PROGRAMMING EXERCISE

문성훈

Copyright 2019

EXERCISE-01

1. 홀수 개의 숫자로 구성된 숫자문자열을 입력으로 사용한다.

※ 숫자문자열의 개수는 7이상 11이하이다.

2. 숫자열의 중앙 숫자를 기준으로 앞과 뒤 두 개의 수로 분리한 후 분리된 두 수를 각각 거꾸로 뒤집은 다음 두 수의 차를 구한다.

[입력 숫자문자열] 7652310

1) 중앙의 수 2를 기준으로 765와 310으로 분리한다.

2) 분리된 두 수를 각각 거꾸로 뒤집는다.

※ 거꾸로 뒤집은 숫자가 “0” 으로 시작할 경우 “0” 을 제거한다.

- 765 => 567

- 310 => 013 => 13

3) 큰 수인 567에서 작은 수인 13을 뺀다.

$567 - 13 = 554$

EXERCISE-02

1. 입력으로 제공된 숫자열을 뒤에서부터 두 자리마다 끊어 읽는다.

※ 입력 숫자열 : 6118266283093

=> 6 | 11 | 82 | 66 | 28 | 30 | 93

2. 끊어 읽은 두 자리가 모두 짝수인 수를 찾아 개수를 계산한다.

※ “0”은 짝수가 아니다.

※ 마지막에 끊어 읽은 수가 두 자리가 아닌 경우는 개수 계산에서 제외한다.

예) 6 | 11 | 82 | 66 | 28 | 30 | 93

X X 0 0 0 X X => 3개

3. 여러 개의 숫자열 목록을 리스트로 사용한다.

예) 6118266283093 20445610 443564831 389593 284281569493 22686448

4. 입력으로 제공된 각각의 숫자열에 대해 1,2번 과정을 진행해 숫자열마다 찾은 짝수의 개수 중 가장 많은 짝수 개수를 구한다.

예) 6118266283093 20445610 443564831 389593 284281569493 22686448

3

1

1

0

2

4

=> 가장 많은 짝수 개수 : 4

EXERCISE-03

1. 입력으로 최대 100자의 문자열이 제공된다.
2. 입력된 문자열에서 숫자만을 추출해서 출력하세요.
3. 추출된 숫자 문자열에서 개수가 가장 많은 수를 찾아 숫자와 출현빈도를 출력하세요
단, 개수가 동일한 경우 작은 수를 선택한다.

예1)

[입력] Hi123987IM100SN123L23457MD31

[숫자 추출] 1239871001232345731

[개수가 가장 많은 숫자] 1(4번)

※ 1 -> 4개, 3 -> 4개로 개수가 동일하므로 작은 수인 1을 선택.

예2)

[입력]: N1235DJ4954G99H87KAU9889EL4234GO312I199

[숫자 출력]: 12354954998798894234312199

[개수가 가장 많은 숫자] 9(7번)

EXERCISE-04

1. 다음과 같은 특성을 갖는 숫자의 개수를 찾는 기능을 구현합니다.

- 입력으로 두개의 숫자(x, y)를 이용합니다.
- 두 개의 숫자 x 와 y 를 이용하여,
 x 초과 y 미만의 숫자 중 각 자리의 숫자를 모두 더한 값이 5의 배수가 되는 숫자를 찾습니다.
- 숫자들을 모두 찾은 후 해당 숫자가 총 몇 개인지를 출력합니다.

예1) 두 개의 숫자 1과 100이 주어졌을 경우,

1초과 100미만의 숫자 중 각 자리의 숫자를 모두 더한 값이 5의 배수가 되는 숫자를 찾습니다.

- 20의 경우 각 자리 숫자를 모두 더한 값이 2이므로, 적합하지 않다.
- 23의 경우 각 자리 숫자를 모두 더한 값이 5이므로, 적합하다.

[총 개수] 19

예2) 두 개의 숫자 5와 500이 주어졌을 경우,

5초과 500미만의 숫자 중 각 자리의 숫자를 모두 더한 값이 5의 배수가 되는 숫자를 찾습니다.

[총 개수] 98

EXERCISE-05

1. 주어진 수의 숫자들 중 홀수를 짝수로 변환한다.

※ 0은 짝수로 취급한다.

※ 홀수인 수에 2를 곱하여 짝수로 만든다.

※ 2를 곱한 결과가 10 이상이라면 일의 자릿수를 선택한다.

[주어진 수] 3278533

[주어진 수의 숫자 중 홀수를 짝수로 변환] 6248066

2. 1번의 결과를 이용하여 앞에서부터 두 자리씩 덧셈 연산을 반복한다.

1) 숫자의 자릿수가 홀수인 경우 마지막 일의 자리는 그대로 사용한다.

2) 두 자리씩 덧셈 연산한 결과가 10 이상이라면 일의 자릿수를 선택한다.

3) 수가 하나가 되어 더 이상 덧셈을 하지 못할 때까지 반복하여 연산한다.

4) 두 자리씩 연산하지 못하는 경우 2.1 방식으로 그대로 다음 연산에 사용한다.

[1번의 결과] 6248066

[두 자리씩 덧셈 연산 예시]

[덧셈 연산 결과] 2

6	2	4	8	0	6	6
8		1 <u>2</u>		6		6
1 <u>0</u>				1 <u>2</u>		
2						

또 다른 예시

입력: 12456984
[덧셈 연산 결과]: 4

EXERCISE-06

1. 6자리 이상 9자리 미만의 수가 입력으로 사용된다.

※ 입력된 수에는 중복된 숫자가 없다.

※ 입력된 수에는 숫자 0은 포함되지 않는다.

2. 수의 중앙을 기준으로 두 개의 수로 분리한 후 큰 수를 선택한다.

※ 수의 숫자개수가 홀수 개인 경우 수의 중앙 숫자를 기준으로 왼쪽과 오른쪽 수로 분리

※ 수의 숫자개수가 짝수 개인 경우 수를 반으로 나누어 왼쪽과 오른쪽 수로 분리

예1) 1234567 → (123, 567) → (567)

예2) 34217869 → (3421, 7869) → (7869)

3. 입력으로 제공된 수를 더 이상 두 개의 수로 분리할 수 없을 때까지 2번의 과정을 반복하여 남은 최종 숫자를 구해 출력한다.

예1) 567 → (5, 7) → (7)

예2) 7869 → (78, 69) → (78) → (7, 8) → (8)