

## <C프로그래밍 및 실습> 6장 반복문 실습 문제

### ※ 문제에 대한 안내

- 출력 예시에서 □는 출력되는 공백을 의미한다.
- 입출력 예시에서 ↳ 이 후는 각 입력과 출력에 대한 설명이다.

[ 문제 9-1 ] 하나의 양의 정수 N을 입력 받아, N에서 숫자 3이 총 몇 번 나타나는지를 출력하는 프로그램 작성하시오.

- N=33 일 때, 2개의 3이 나타난다.

입력 예시 1	출력 예시 1
131	1
입력 예시 2	출력 예시 2
303	2
입력 예시 3	출력 예시 3
31323	3

6장5절 [ 문제 9 ] 하나의 양의 정수 N을 입력 받아, 1부터 N까지의 정수에서 숫자 3이 총 몇 번 나타나는지를 출력하는 프로그램 작성하시오.

- 중첩 반복문 (for 문, while 문 사용)
- N=33 일 때, 3, 13, 23, 30, 31, 32, 33, 총 7개의 수에서 8개의 3이 나타난다.  
(∵ 숫자 33에는 3이 두 개 있으므로)

입력 예시 1	출력 예시 1
10	1
입력 예시 2	출력 예시 2
33	8
입력 예시 3	출력 예시 3
31323	13711

[ 문제 10-1 ] 10이상의 양의 정수 N을 입력 받아 각 자리수의 합을 출력하는 프로그램을 작성하시오.

- 예) 6234 → 6+2+3+4=15

입력 예시 1	출력 예시 1
6234	15
입력 예시 2	출력 예시 2
123456789	45
입력 예시 3	출력 예시 3
40	4

6장5절 [ 문제 10 ] 10이상의 양의 정수 N을 입력 받아 디지털근을 출력하는 프로그램을 작성하시오. N의 각 자리수를 더한다. 한 자리 수가 될 때까지 반복한다.

- 디지털근, 예)  $6234 \rightarrow 6+2+3+4 = 15 \rightarrow 1+5 = 6$   
 예)  $52561 \rightarrow 5+2+5+6+1 = 19 \rightarrow 1+9 = 10 \rightarrow 1+0 = 1$
- 중첩 while 문 이용해야 해결됩니다.

입력 예시 1	출력 예시 1
6234	6
입력 예시 2	출력 예시 2
123456789	9
입력 예시 3	출력 예시 3
40	4

6장5절 [ 문제 11 ] 3~18사이의 정수 N을 입력 받아, 3개의 주사위의 총 합이 N이 되는 모든 경우를 출력하는 프로그램을 작성하시오.

- 중첩 반복문
- 각 줄의 수는 하나의 공백을 사이에 두고 출력, 맨 마지막에는 공백 없음  
 $1 \square 3 \square 6$

입력 예시	출력 예시
10	1 3 6 1 4 5 1 5 4 1 6 3 2 2 6 2 3 5 2 4 4 2 5 3 2 6 2 3 1 6 3 2 5 3 3 4 3 4 3 3 5 2 3 6 1 4 1 5 4 2 4 4 3 3 4 4 2 4 5 1 5 1 4 5 2 3 5 3 2 5 4 1 6 1 3 6 2 2 6 3 1

6장6절 [ 문제 12 ] 하나의 양의 정수 N을 입력 받아 1부터 N까지의 합을 출력하는 프로그램을 작성하시오.

- for 문의 무한루프 이용
- break 문 이용

입력 예시 1	출력 예시 1
10	55
입력 예시 2	출력 예시 2
33	561
입력 예시 3	출력 예시 3
40	820

6장7절 [ 문제 13 ] 정수 N을 반복해서(종료 조건 시까지) 입력받아 N의 합계, 평균, 개수를 출력하는 프로그램을 작성하시오.

- 종료 조건: 0 이하 정수 입력 (합계, 평균, 개수 계산에는 제외)
- 입력된 수 중에서 100을 넘는(초과하는) 수는 계산에서 제외 한다 (예시2를 보시오)
- while 문, if 문, break 문, continue 문 이용
- 실수는 소수점 이하 둘째 자리까지만 출력하시오.

입력 예시 1	출력 예시 1
2 8 3 7 4 6 -2	30 5.00 6    ↳ 합계, 평균, 개수
입력 예시 2	출력 예시 2
1 4 101 200 -10	<u>5</u> <u>2.50</u> 2
입력 예시 3	출력 예시 3
10 20 30 40 50 60 70 0	280 40.00 7

## 7주차 실습문제

[ 문제 1 ] 양의 정수 N을 입력 받아, N의 자리수를 역순으로 바꾼 정수 M을 구하여 출력하고, 1부터 M까지의 정수 중 2의 배수의 개수와 3의 배수의 개수를 출력하는 프로그램을 작성하시오.

- 만일 2의 배수나 3의 배수가 없으면 0을 출력한다.

입력 예시 1	출력 예시 1
12            ↳ 정수 N	21            ↳ 역순으로 바꾼 정수 M
	10 7            ↳ M의 2의 배수 개수와 3의 배수 개수

1부터 자리수를 역순으로 바꾼 정수인 21까지의 2의 배수 개수, 3의 배수 개수 출력

입력 예시 2

출력 예시 1

10	↦ 정수 N	1	↦ 역순으로 바꾼 정수 M
		0 0	↦ M의 2의 배수 개수와 3의 배수 개수

**[ 문제 2 ] 종료조건(0 입력)까지** 양의 정수 N을 반복해서 입력 받는다. 입력 받은 정수 N 뒤에, N의 자리수를 역순으로 바꾼 정수를 붙여서, 대칭이 되는 정수 M을 만들어 출력하고, 대칭이 되는 정수 M의 각 자리수에서 3이 나오는 개수를 출력한다. 또한, 대칭되는 정수들 중 최소값과 최대값을 찾아 출력하는 프로그램을 작성 하시오.

- 배열 사용 금지 (위반 시 100% 감점)

입력 예시 1

출력 예시 1

13 42 122 0	↦ 정수	1331 2 4224 0 122221 0
		↦ 대칭정수 대칭정수에 포함된 3의 개수
		1331 122221 ↦ 최소값 최대값

1331에서 3이 두 개이므로 1331과 2를 출력한다.

입력 예시 2

출력 예시 2

1 5000 32 0	↦ 정수	11 0 50000005 0 3223 2
		↦ 대칭정수 대칭정수에 포함된 3의 개수
		11 50000005

**[ 문제 3 ]** 정수 N을 입력받고, N개의 양의 정수를 입력받아, 입력된 정수 중 소수만을 한 줄에 3개씩 출력하는 프로그램을 작성하시오.

- 마지막 줄에 출력되는 소수의 개수는 3개보다 작을 수 있다.
- 각 줄 마지막에 별표(\*)를 출력하시오.
- N개의 양의 정수 중, 소수가 최소 1개 이상이라고 가정한다.

입력 예시 1

출력 예시 1

11	↦ N=11	2 3 5 *
1 2 3 4 5 6 7 8 9 10 11		7 11 * ↦ 마지막 줄엔 소수 2개만 출력

입력 예시 2

출력 예시 2

15	↦ N=15	11 13 17 *
11 13 17 19 71 73 74 2 3 5 58 97 89 79 45		19 71 73 *
		2 3 5 *
		97 89 79 *

**[ 문제 4 ]** (정수 N을 입력받고, N개의 양의 정수를 입력받아, 입력된 정수 중 소수만을 출력한다. 출력 방식은 연속적으로 소수가 나오는 경우 같은 줄에 출력하고, 소수가 아닌 정수가 입력되다가 다시 소수가 나오면 줄을 바꾸어 출력하도록 프로그램을 작성하시오.

- 연속적으로 나오는 소수는 개수에 관계없이 모두 한 줄에 출력한다.
- 각 줄 마지막에 별표(\*)를 출력하시오.
- N개의 양의 정수 중, 소수가 최소 1개 이상이라고 가정한다.

입력 예시 1	출력 예시 1
15 $\mapsto$ N=15 11 13 17 19 71 73 74 2 3 5 58 97 89 79 45	11 13 17 19 71 73 * 2 3 5 * 97 89 79 *
입력 예시 2	출력 예시 2
12 $\mapsto$ N=12 71 73 74 2 4 6 58 97 89 79 11 37	71 73 * 2 * 97 89 79 11 37 *

**[ 문제 5 ]** 양의 정수 N과 M을 입력받고, N개의 양의 정수를 입력받아 이 중 M번째 소수까지만 출력하는 프로그램을 작성하시오.

- 만일 M개의 소수보다 적으면 있는 소수를 모두 출력하고, 소수가 하나도 없으면 0을 출력한다.

입력 예시 1	출력 예시 1
10 3 $\mapsto$ 정수 10개를 읽어 소수 3개 출력 1 2 3 4 5 6 7 8 9 10	<input type="checkbox"/> 2 3 5
입력 예시 2	출력 예시 2
10 5 1 2 3 4 5 6 7 8 9 10	<input type="checkbox"/> 2 3 5 7
입력 예시 3	출력 예시 3
10 5 32 33 34 35 36 42 44 45 46 54	0

**[ 문제 6 ]** 양의 정수 N과 N개의 양의 정수를 입력받아 연속적으로 소수가 나온 최대 회수를 카운트하고 그 구간의 마지막 소수를 출력 하는 프로그램을 작성하시오.

- 단 100 이상인 수는 제외 한다 (소수로 생각하지 않음).
- 소수가 6번 연속 나온 구간이 제일 길다면 6을 출력한다. (입력 예시 1)
- 입력 중간에 100이상의 소수가 나와도 100이상의 수이므로 소수 구간은 새로 시작해야한다. (입력 예시 2)
- 소수연속구간 최대 회수를 출력하고, 다음 줄에 그 구간의 소수들 중 가장 마지막 소수를 출력한다.
- 한 번도 소수가 없었다면 0을 출력한다. (입력 예시 3)
- 최대 길이가 같은 구간이 있으면 제일 처음 나온 구간의 정보를 출력한다.

입력 예시 1

15  
11 13 17 19 71 73 74 2 3 5 58 97 89 79 47

출력 예시 1

6 ↦ 11 13 17 19 71 73  
73 ↦ 그 구간의 마지막 소수

입력 예시 2

17  
11 13 17 19 101 71 73 74 2 3 5 58 97 89 79 11 37

출력 예시 2

5 ↦ 97 89 79 11 37  
37 ↦ 그 구간의 마지막 소수

입력 예시 3

10  
32 33 34 35 36 42 44 45 46 54

출력 예시 3

0

**[ 문제 7 ]** 영문 소문자 C1, C2와 영문 소문자 세 개로 이루어진 단어 T를 C1C2-T의 형식으로 입력받아,

- (1) C1과 C2를 포함한 C1과 C2 사이의 소문자 세 개로 이루어진 단어 중 단어 생성 조건을 만족하는 단어를 사전에 나오는 순서로 출력한다.

※ 단어 생성 조건

- (1) 세 문자가 모두 같은 문자인 경우는 없다. 즉, 세 문자는 모두 다른 문자이거나, 세 문자 중 두 문자만 같은 문자일 수 있다.  
(2) **두 번째 문자는 모음('a', 'e', 'i', 'o', 'u' 중 하나)이다.**  
(3) 두 번째 문자는 세 문자의 중간값이다. 즉, 세 문자를 알파벳 순서로 줄을 세운 경우, 두 번째 문자가 중간 순서에 위치한다. 단, **aab 또는 baa처럼 같은 문자가 중복된 경우에도 알파벳 순서로 줄을 세우면 a-a-b순으로 문자 a가 중간값이 된다.**

(2) (1)에서 출력된 단어 중에서, 단어 T보다 사전순서상 빠른 가장 마지막으로 출력된 단어를 출력한다.

- (1)의 출력 순서는 단어가 사전에 나오는 순서이다. 사전에 나오는 순서란 알파벳 순서가 빠른 문자로 이루어진 단어가 먼저 출력된다는 의미이다. 예를 들어, aab와 abc를 비교해보면, 첫 번째 문자는 'a'로 동일하지만, 두 번째 문자인 aab의 'a'가 abc의 'b'보다 빠르기 때문에 aab가 abc보다 사전 순서 상 빠른 단어이다.
- C1과 C2 사이의 문자로 조건에 맞는 단어가 만들어지지 않는 경우에는 "none"을 출력한다.
- 사전순서상 T보다 빠른 단어를 찾을 수 없는 경우에도 "none"을 출력한다. (출력 예시 2와 3 참고)
- C1은 C2 보다 알파벳 순서 상 빠른 문자라고 가정한다.

입력 예시 1

dg-fff  
↦ C1C2-T 형태 입력  
↦ C1은 'd', C2는 'g'  
↦ T는 "fff"

출력 예시 1

dee ↦ 'd', 'e', 'f', 'g'로 만들 수 있는 64개의 단어 중 조건을 만족하는 단어만 출력  
def  
deg  
eed  
eef  
eeg  
fed  
fee  
ged  
gee  
fee ↦ 출력된 단어 중 "fff"보다 빠른 가장 마지막 단어 출력

입력 예시 2

출력 예시 2

np-nno	noo	↳ 'n', 'o', 'p'로 만들 수 있는 27개의 단어 중 조건을 만족하는 단어만 출력
↳ C1C2-T 형태 입력	nop	
↳ C1은 'n', C2는 'p'	oon	
↳ T는 "nno"	oop	
	pon	
	poo	
	none	↳ 출력된 단어 중 "nno"보다 빠른 단어가 없어서 "none" 출력

입력 예시 3

출력 예시 3

jn-mmm	none	↳ 'j'에서 'n' 사이에는 모음이 존재하지 않아 조건에 맞는 단어가 만들어지지 않아 "none" 출력
	none	↳ 출력된 단어 중 "mmm"보다 빠른 단어가 없어서 "none" 출력

[ 문제 8 ] 세 학과가 공동으로 사용하는 건물의 출입을 위한 비밀번호는 학과 학생들이 개별 설정할 수 있다. 단, 비밀번호는 다음의 규칙에 의해 생성된다.

※ 비밀번호 생성규칙

- (1) 다섯 자리 정수(10000에서 99999사이의 정수)를 사용한다.
- (2) 마지막 두 자리의 숫자는 00이다.
- (3) 세 번째 자리는 학과 코드이며, 학과별로 다음과 같다.

코드	0	1	2	3	4	5
학과명	CE	CE	CE	SW	SW	DS

- (4) 첫 번째(C1), 두 번째(C2), 세 번째(C3) 자리의 숫자는 모두 같지 않고, 모두 다르지도 않다. 즉, 세 개의 숫자 중, 두 개는 같은 숫자이고, 나머지 하나는 다른 숫자이다.

예: [입력] 11300에 대해서 C1:1, C2:1, C3:3

정수 N을 입력받고, N명 학생이 건물 출입을 위해 입력한 비밀번호를 입력받아, 각 비밀번호를 입력한 학생의 학과명을 출력하는 프로그램을 작성시오. 단, 비밀번호 생성규칙에 맞지 않는 입력에 대해서는 "none"을 출력한다.

입력 예시 1

출력 예시 1

11	↳ 정수 N, 입력갯수	SW
11300	↳ 유효비밀번호(학과코드 3)	CE
20000	↳ 유효비밀번호(학과코드 0)	none
310100	↳ 입력오류(6자리 정수)	CE
10100	↳ 유효비밀번호(학과코드 1)	CE
21200	↳ 유효비밀번호(학과코드 2)	none
11600	↳ 입력오류(존재하지 않는 학과코드)	none
11510	↳ 입력오류(마지막 두 자리가 00이 아님)	none
10200	↳ 입력오류(처음 세 자리가 모두 다른 수)	SW
33400	↳ 유효비밀번호(학과코드4)	DS
75500	↳ 유효비밀번호(학과코드5)	none
1100	↳ 입력오류(4자리 정수)	