

## 프로그래밍 과제 03: 1, 3, 5, 7, 8번

(모든 문제에 대해서 배열, 중첩된 반복문, stdio.h 이외의 라이브러리를 사용해서는 안된다.)

- 키보드로 부터 양의 정수들을 연속해서 입력받는다. 입력으로 들어온 홀수의 합과 짝수의 합이 동일해지면 그때까지 입력된 정수의 개수를 출력하고 종료한다. 단, 적어도 하나 이상의 입력을 받아야 한다.

입력 예	출력
2 1 1	3
1 2 2 5 2	5
2 3 1 2	4
3 2 1 1 1 4	6

- 입력으로 하나의 양의 정수  $N$ 을 받은 후 이진수로 변환하여 출력하는 프로그램을 작성하라. 프로그램의 골격은 다음과 같다.

```
#include <stdio.h>
int main(void) {
    int N;
    scanf("%d", &N);

    /* 여기에서 N보다 작거나 같으면서 가장 큰 2의 거듭제곱수 v를 구한다. */

    while (v > 0) {
        /* N ≥ v이면 N = N-v가되고 */
        /* 1을 출력한다. 그렇지 않으면 */
        /* 0을 출력한다. 두 경우 모두 */
        /* v는 1/2한다. */
    }
}
```

입력 예	출력
123	1111011
1111111	100001111010001000111
1024	10000000000
0	0

- 입력으로 두 양의 정수를 받은 후 두 정수의 최대공약수(GCD)를 구해서 출력하는 프로그램을 작성하라. GCD를 구하기 위해서 Euclid 알고리즘을 사용하라. Euclid 알고리즘은 다음의 성질을 이용한다: 두 정수  $x, y$ 중에 크거나 같은 쪽을  $x$ 라고 하자. 만약  $x$ 가  $y$ 로 나누어 떨어지면 GCD는  $y$ 이다. 그렇지 않다면  $x$ 와  $y$ 의 GCD는  $x \% y$ 와  $y$ 의 GCD와 같다.

입력 예	출력
24 36	12
1234 5678	2
248 384	8
16 816	16

4. 사용자로부터 하나의 양의 정수  $n$ 을 입력 받은 후 자리를 뒤집어서 출력하는 프로그램을 작성하라. 예를 들어  $n = 12345$ 이면 54321을 출력하면 된다.
5. 입력으로 -1이 들어올 때 까지 연속해서 정수들을 입력받는다. -1이 입력되면 그때까지 입력된 정수들 중에서 가장 큰 수와 2번째로 큰 수를 출력하라. (-1 자체는 입력된 정수로 간주하지 않는다.)

입력 예	출력
-5 0 9 2 3 8 -7 -1	9 8
7 2 3 4 8 1 8 4 0 2 -1	8 8
-5 -6 -9 -2 -7 -19 -6 -1	-2 -5
2 2 2 2 2 2 -1	2 2

6. Newton's method를 이용하여 제곱근을 구하는 프로그램을 작성하라. 가령  $x$ 의 제곱근을 구할 때 Newton's method는 일단 하나의 초기값  $y$ 에서 시작하는데 보통  $y = 1$ 을 택한다. 그다음  $y$ 와  $x/y$ 의 평균을 계산하여 이것을  $y$ 의 새로운 값으로 선택한다. 이 과정을  $y$ 와 새로운  $y$ 값의 차이가 오차범위  $\epsilon$  이내가 될 때까지 반복한다.  $\epsilon$ 은 0.00001 곱하기 그 시점의 새로운  $y$ 의 값으로 하라. 예를 들어 3의 제곱근을 구하는 과정은 다음과 같다.

X	Y	X/Y	Y와 X/Y의 평균 (새로운 Y값)	$\epsilon$
3	1	3	2	0.00002
3	2	1.5	1.75	0.0000175
3	1.75	1.71429	1.73214	0.0000173214
3	1.73214	1.73196	1.73205	0.0000173205
3	1.73205	1.73205	1.73205	0.0000173205

7. 키보드로부터 2개 이상의 정수들을 연속해서 입력받는다. -1은 입력의 끝을 의미한다 (즉 -1 자체는 입력된 정수로 간주하지 않는다.) 입력된 정수들 중에서 local minima의 개수를 카운트하여 출력하라. 여기서 local minima란 바로 직전에 입력된 정수보다 작거나 같고, 바로 다음에 입력된 정수보다도 작거나 같은 정수를 의미한다. 단, 첫 번째 정수는 2번째 정수보다 작거나 같으면 local minima로 간주하고, 마지막 정수는 바로 이전의 정수보다 작거나 같으면 local minima로 간주한다.

입력 예	출력
1 1 2 1 3 5 9 8 -1	4
1 2 -1	1
9 7 5 7 8 1 3 9 -1	2
1 2 5 4 5 2 7 8 1 1 1 1 2 -1	7

8. 아래의 표는 우리 나라의 소득 구간별 근로소득세율이다. 가령 소득이 1억원이면 소득세는 다음과 같이 계산된다.

소득	세율
1,200만원 이하	6%
1,200만원 초과 4,600만원 이하	15%
4,600만원 초과 8,800만원 이하	24%

소득	세율
8,800만원 초과 1.5억원 이하	35%
1.5억원 초과 5억원 이하	38%
5억원 초과	40%

$12,000,000 * 6\% = 720,000\text{원}$ . (1억원 중 12,000,000원에 대해서는 6%)  
 $34,000,000 * 15\% = 5,100,000\text{원}$  (1억원 중 1200만원초과 4600만원 이하인 3400만원에 대해서는 15%)  
 $42,000,000 * 24\% = 10,080,000\text{원}$  (1억원 중 4600만원 초과 8800만원 이하인 4200만원에 대해서는 24%)  
 $12,000,000 * 35\% = 4,200,000\text{원}$  (1억원 중 나머지 1200만원에 대해서는 35%)  
**총합: 20,100,000원**

소득 금액을 입력으로 받아서 소득세를 계산하여 출력하는 프로그램을 작성하라.

입력 예	출력
10000000	600000
20000000	1920000
30000000	3420000
50000000	6780000
100000000	20100000
200000000	56600000
600000000	210600000

9. 어떤 수열에 대해서 어떤 수도 자신의 이전 값보다 크거나 같을 때 비감소(non-decreasing) 수열이라고 부른다. 반대로 어떤 수도 자신의 이전 값보다 작거나 같을 때 비증가(non-increasing) 수열이라고 부른다. 단, 길이가 1인 수열은 비감소 수열이면서 동시에 비증가 수열이다. 하나의 비감소 수열 뒤에 다시 하나의 비증가 수열이 연결된 형태의 수열을 볼록(convex) 수열이라고 부른다. 예를 들어 1 1 2 3 3 4 4 3 1 1은 볼록 수열이다. 또한 1 2 3 4 5 역시 볼록 수열이다. 반대로 하나의 비증가 수열 뒤에 하나의 비감소 수열이 연결된 형태의 수열을 오목(concave) 수열이라고 부른다. 하나의 수열을 받은 후 이것이 볼록 수열인지, 오목 수열인지, 둘 다인지, 혹은 둘 다 아닌지 판단하는 프로그램을 작성하라. 입력에서 -1은 수열의 끝을 의미한다 (즉 -1 자체는 수열의 일부가 아니다.)

입력 예	출력
1 2 3 4 5 -1	Both
1 1 1 3 3 3 3 3 2 2 2 2 -1	Convex
2 2 2 2 2 2 -1	Both
3 3 3 3 2 2 2 2 1 1 -1	Both
1 1 1 3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6 6 6 1 1 -1	Convex
1 1 1 1 3 3 3 3 4 4 4 4 3 3 3 3 4 4 4 -1	None
5 5 2 2 2 2 3 3 3 3 2 2 2 2 -1	None
5 5 5 5 4 4 4 4 2 2 2 2 2 3 3 3 3 9 9 9 9 -1	Concave