

2-1. (2의 배수 여부 판별)(난이도 하)(연습)

성용이는 임의의 자연수를 다른 숫자로 변경하는 함수를 생각하고 있다. 성용이가 원하는 함수는 입력한 자연수를 2의 배수인지 확인하여 2의 배수이면 1로 아니면 0으로 표시하는 것이다. 예를 들어 입력한 숫자가 21인 경우에는 2의 배수는 아니므로 0으로 20의 경우에는 2의 배수이므로 1로 표시한다. 이처럼 입력한 숫자에 대해서 2의 배수인지 아닌지를 1과 0으로 출력하는 프로그램을 작성하라. 입력하는 숫자는 0보다 크며 10^{51} 보다는 작거나 같다.

Hint) 2의 배수는 입력 받은 숫자의 1의 자리 수가 짝수이면 2의 배수이다.

Hint2) 큰 수를 다루려면 문자열을 사용해야한다.

(입력) 1355

(출력) 0 <- 2의 배수 아님

(입력) 11111111111121

(출력) 0 <- 2의 배수 아님

(입력) 3062718240012612151510011152120

(출력) 1 <- 2의 배수임

2-2. (3,4,5,6배수 여부로 인코딩)(난이도 중하)(프로그래머스 제출)

이번 문제에서는 2/3/4/5의 배수인지 확인하여 각 숫자의 배수이면 차례로 1로 표시하고 그렇지 않으면 0으로 표시하고자 한다.

예를 들어 입력한 숫자가 21인 경우에는 2의 배수는 아니므로 0, 3의 배수이므로 1, 4와 5의 배수가 아니므로 각각 0을 차례로 표시한다. 즉, 그 결과를 0100으로 표시한다. 또 다른 입력한 숫자가 18이면 18은 2와 3의 배수이므로 각각 1로, 4와 5의 배수가 아니므로 각각 0을 차례로 표시하여 1100이다. 이와 같이 출력해 주는 프로그램을 작성하시오. 입력하는 숫자는 0보다 크며 10^{51} 보다는 작거나 같다.

Hint)

(1) 2의 배수: 입력 받은 숫자의 1의 자리 수가 짝수이면 2의 배수이다

(2) 5의 배수: 1의 자리 수가 0또는 5이면 5의 배수이다.

(3) 3의 배수: 입력 받은 수의 각 자리 수의 합이 3의 배수이면 3의 배수이다.

(4) 4의 배수: 입력 받은 숫자를 100으로 나눈 나머지가 4로 나누어지면 4의 배수이다. 예를 들어 입력한 숫자가 1256이면 56이 4의 배수이므로 1256은 4의 배수이다.

(입력) 1355

(출력) 0001 → 2/3/4의 배수가 아니지만 5의 배수임.

(입력) 11111111111121

(출력) 0100 → 2/4/5의 배수가 아니지만 3의 배수임.

(입력) 3062718240012612151510011152120

(출력) 1011 → 2/4/5의 배수지만 3의 배수는 아님.

3-1. (자리수 구하기)(난이도 하)(연습)

자연수(21억보다 작은)인 임의의 수 n 을 입력받아 그 자리수를 반환하는 함수 `solution`을 작성하라. 함수의 원형은 다음과 같다.

```
int solution(int n);
```

매개변수 n 이 63일 때 자리수인 2를 반환한다.

매개변수 n 이 39399일 때 자리수인 5를 반환한다.

3-2. (“3의 배수” 게임)(난이도 상)(프로그래머스) 흥민이는 “3의 배수”라는 새로운 게임을 만들어서 친구들과 여가 시간을 보내기로 했다. 게임의 룰은 다음과 같다.

- (1) 숫자를 3부터 시작해서 3의 배수를 차례로 이야기한다. 첫 번째 사람은 3, 두 번째 사람은 6, 세 번째 사람은 9를 말한다.
- (2) 10보다 큰 3의 배수는 한 자리씩 끊어서 말한다. 네 번째 사람은 1, 다섯 번째 사람은 2, 여섯 번째 사람은 다시 1, 일곱 번째 사람은 5를 말한다.

이런 순서로 게임을 진행하면

3 6 9 1 2 1 5 1 8 2 1 2 4 2 7 3 0 ...

의 순서로 숫자를 말하게 된다.

이렇게 숫자를 외치면서 즐거운 시간을 보내다가 문득 N 번째 사람은 어떤 숫자를 말하게 될지 궁금해졌다. 이것을 계산하여 출력해 주는 프로그램을 작성하시오. 입력하는 숫자 N 은 자연수이고 10만보다는 작다.

(입력) 10

(출력) 2

(입력) 3

(출력) 9

(입력) 15

(출력) 7

힌트: 위에서 작성한 `solution` 함수를 사용하여도 좋다(물론 안 사용해도 된다).

4-1. (뽑기-일정합 수식 존재 여부)(난이도 중)(프로그래머스)1부터 n까지 연속되어 있는 수와 +/-를 이용하여 만들 수 있는 모든 수식 중에서 추가로 입력한 숫자(N)로 계산되는 결과가 있으면 1 그렇지 않으면 0을 출력하는 프로그램을 작성하시오. 입력은 n, N이다.

입력) 2 3 (→ n이 2이고 N이 3임)

출력) 1

1부터 2까지 연속된 수와 +/-를 이용해서 만들 수 있는 수식은 아래와 같으며

$$+1+2 = 3$$

$$+1-2 = -1$$

$$-1+2 = 1$$

$$-1-2 = -3$$

이 중 3으로 계산되는 경우가 있으므로 1을 출력한다.

입력) 2 0 (→ n이 2이고 N이 0임)

출력) 0

위 해설과 같이 0으로 계산되는 경우가 없으므로 0을 출력한다.