1. 두 개씩 곱하기 (4점)

6개의 수 1, 2, 3, 4, 5, 6이 있다.

이들 중 서로 다른 두 수 x, y $(x < y)$를 골라서 곱한 값들을 모두 더하면?

» 21

» 91

» 175

» 350

» 441

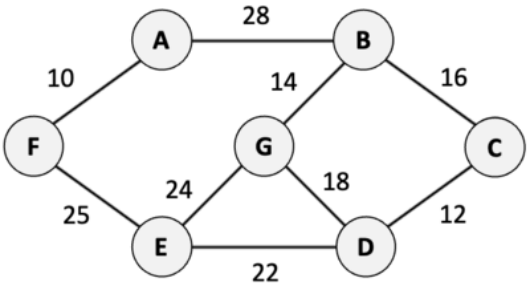
2. 진법 변환 (5점)

16진수 `EA`를 8진수로 표현하면 `352`이며, 이 때 나타나는 각 숫자의 합은 $3 + 5 + 2 = 10$ 이다.

16진수 `A6E4F3`을 8진수로 표현했을 때 나타나는 각 숫자의 합은?

3. 최소 신장 트리 (5점)

다음 그래프의 최소 신장 트리의 가중치는? 트리의 가중치는 트리에 포함된 간선의 가중치의 합이다.



4. 기사, 도둑, 바보 (5점)

항상 진실만 이야기하는 기사, 항상 거짓만 이야기하는 도둑, 진실을 이야기할 수도 거짓을 이야기할 수도 있는 바보 세 가지 종류의 사람이 사는 섬이 있다.

이 섬에 사는 세 사람 A, B, C가 다음과 같이 이야기했다.

- A: 우리 중 한 명 이상이 바보이다.
- B: A는 바보가 아니다.
- C: 나는 도둑이다.

세 사람에게 기사, 도둑, 바보의 역할을 각각 배정하는 서로 다른 경우의 수는 $3^3 = 27$가지이다.

이 중에서, A, B, C가 한 말과 모순이 발생하지 않도록 역할을 배정하는 경우의 수는 몇 가지인가?

5. 참으로 만들기 (5점)

p, q, r, s는 참 또는 거짓의 값을 가질 수 있는 변수이다.

이 4개의 변수에 참 또는 거짓을 배정하는 경우의 수는 $2^4 = 16$가지이다.

이 중에, 식 $((p \vee q) \longrightarrow (\neg q \vee r)) \longrightarrow ((\neg p \vee s) \wedge \neg r)$ 의 값이 참이 되도록 하는 경우의 수는 몇 가지인가?

$\vee$는 OR 연산자, $\wedge$는 AND 연산자, $\neg$는 NOT 연산자이다.

6. 인버전의 개수 (8점)

1, 2, 3, 4, 5, 6, 7, 8이 한 번씩만 나타나는 수열 a_1, a_2, $\cdots$, a_8 이 있다.

c_i 를 $j < i$ 이면서 $a_j > a_i$ 인 j 의 개수라고 하자. c_1, c_2, $\cdots$, c_8 은 아래 표와 같다.

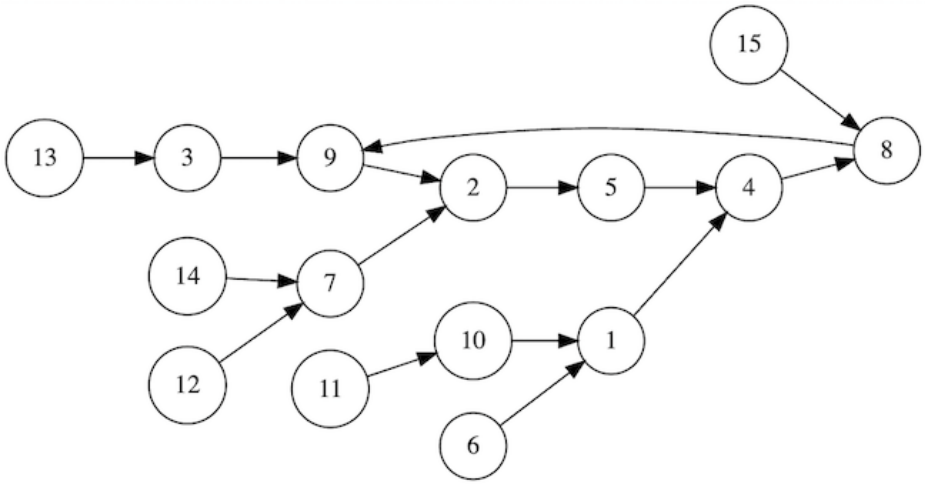
i	1	2	3	4	5	6	7	8
c_i	0	1	0	3	0	5	2	3

위의 정보를 통해 수열 a를 유일하게 결정할 수 있다.

이 때, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 을 공백 없이 순서대로 나열한 8자리 문자열을 구하라.

7. 화살표 따라가기 (8점)

아래와 같이 정점이 15개이고, 모든 정점에서 출발하는 간선이 정확히 1개씩 있는 방향 그래프가 있다. 정점에는 1부터 15까지의 번호가 붙어 있다.



$f(u, k)$를, u번 정점에서 시작하여 간선을 k개 거쳤을 때 도달하는 정점의 번호라고 하자. 예를 들어 $f(1, 4) = 2$, $f(1, 7) = 8$, $f(15, 2) = 9$ 이다.

$f(v, 101)$이 5 또는 9인 모든 v의 합은?

8. 비트 연산의 합 (9점)

A 는 0, 1, 2, 3, 4, 5, 6, 7 중 하나이다.

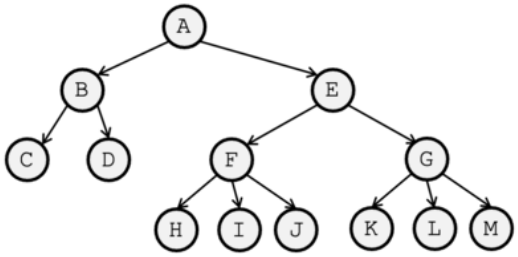
B 는 0, 1, 2, 3, 4, 5, 6, 7 중 하나이다.

$\diamond$ 는 AND, OR, XOR 중 하나이다. (C, C++, Java, Python 언어에서의 `&`, `|`, `^` 비트 연산자와 같다)

모든 가능한 $8 \times 8 \times 3 = 192$가지의 $(A, B, \diamond)$ 순서쌍에 대해, $A \diamond B$ 의 값의 합은?

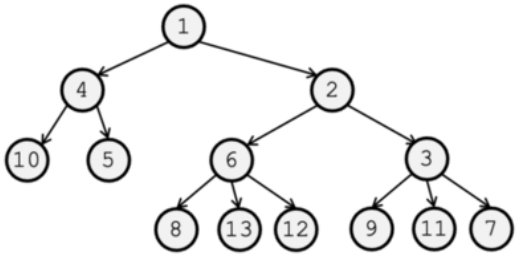
9. 최소 힙 조건 (9점)

아래와 같이 정점이 13개인 트리가 있다.



트리의 각 정점에 1부터 13까지의 서로 다른 자연수를 하나씩 배정하려고 한다. 이 때, 부모 노드에 배정된 수가 자식 노드에 배정된 수보다 반드시 작아야 한다.

예를 들어, 아래와 같은 배정은 유효한 배정이다.



주어진 조건을 만족하도록 자연수를 배정할 수 있는 경우의 수는 $12!/k$ 가지이다. k는 자연수이다. k의 값은?

10. 동전 (11점)

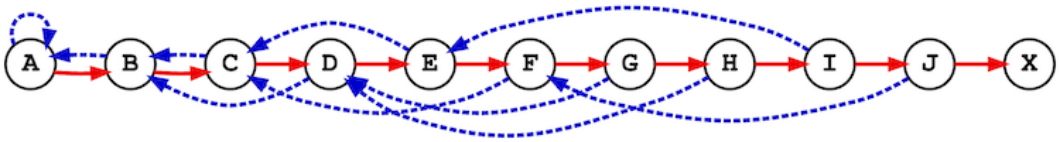
KOI 나라에는 4원, 7원 두 가지 종류의 동전이 있다.

당신은 이 동전으로 물건 값을 내려고 하는데, 이 과정에서 **거스름돈**을 쓸 수 있다. 예를 들어, 10원짜리 물건 값을 내기 위해서 7원 동전 2개를 내고 4원 동전 1개를 거슬러 받을 수 있다.

다만, 이 과정에서 사용할 수 있는 동전의 개수는 당신이 처음 내는 동전과 거슬러 받는 동전을 포함해서 최대 6개로 제한된다. 위 예시에서는 3개의 동전을 사용했다.

1 이상 42 이하의 자연수 n 가운데, 위의 방식대로 n원을 낼 수 있는 경우는 몇 가지인가?

11. 로봇과 그래프 (15점)



위와 같은 방향 그래프가 있다. 어떤 로봇이 정점 A에 놓여 있으며, 정점 X에 도착하는 것이 목표이다.

빨간색 간선들은 왼쪽에서 오른쪽으로 향하며, 실선 형태 (→)로 표시되어 있다. 파란색 간선들은 A A 간선을 제외하고는 오른쪽에서 왼쪽으로 향하며, 점선 형태 ()로 표시되어 있다.

X를 제외한 모든 정점에는 해당 정점에서 출발하는 빨간색 간선과 파란색 간선이 정확히 하나씩 있다.

이때, 로봇은 아래와 같은 방식으로 그래프를 돌아다닐 것이다.

- 현재 있는 정점의 홀수 번째 방문이면 파란색 간선을, 짝수 번째 방문이면 빨간색 간선을 통해 이동한다.

예를 들어 처음 몇 번의 이동 과정에서 방문하는 정점들을 정점(방문 횟수) 형태로 순서대로 표현하면 다음과 같다: A(1) A(2) → B(1) A(3) A(4) → B(2) → C(1) B(3) ...

로봇은 정점 X에 도착할 때까지 총 몇 번 이동하는가? (이동 횟수는 **간선**을 지나간 총 횟수임에 유의하라.)

12. 부분집합의 합 (18점)

게임 속에 서로 다른 아이템이 n개 있다. 각각의 아이템에는 1 이상 8 이하의 정수 **가치**가 있으며, 가치가 i인 아이템은 c_i개 있다. 따라서, $n = c_1 + c_2 + \cdots + c_8$이다.

모든 i $(1 \leq i \leq 8)$에 대해 c_i는 0 이상 9 이하의 정수이다.

아이템들의 부분집합은 총 2^n개 있다. 이 중에, 가치의 합이 정확히 k인 서로 다른 부분집합의 개수를 s_k라고 하자. 공집합의 가치의 합은 0으로 간주한다.

$k = 0, 1, 2, \cdots, 8$에 대해, s_k의 값은 아래 표와 같다.

k	0	1	2	3	4	5	6	7	8
s_k	1	4	9	22	47	86	167	307	525

위의 정보를 바탕으로 c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8을 계산할 수 있다.

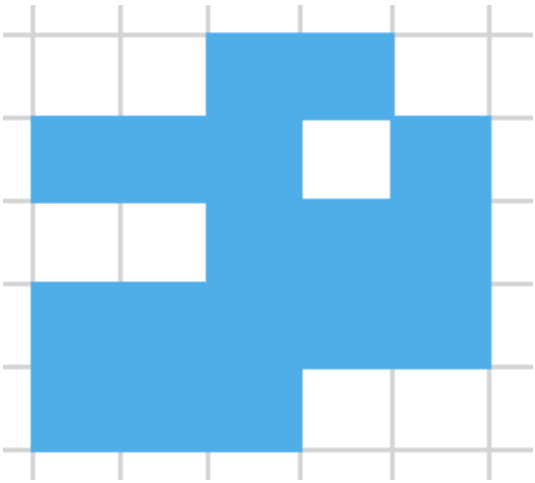
c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8을 공백 없이 순서대로 나열한 8자리 문자열을 구하라.

13. 스탬프 (8점)

아래와 같이 8행 13열의 격자가 있다. 각 격자칸에는 음이 아닌 정수가 적혀 있다.

여러분은 다음 행동을 적절히 수행하여 격자칸에 적힌 모든 수가 0이 되도록 해야 한다.

- 아래 그림과 같은 스탬프를 칸에 맞추어 놓은 후, 스탬프가 놓인 격자칸들에 적힌 값을 모두 1 뺀다.



스탬프를 뒤집거나 돌릴 수 없으며, 스탬프가 격자 밖으로 나가서는 안 됨에 유의하라.

어떤 격자칸을 클릭하면 스탬프의 정가운데 칸을 해당 격자칸에 놓고 행동을 한 번 수행한다. 스탬프가 격자 바깥으로 나가거나, 행동을 수행했을 때 어떤 격자칸에 적힌 값이 음수가 되면, 클릭을 할 수 없다.

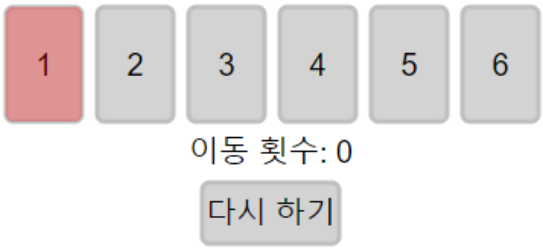
0	0	0	0	0	4	5	5	4	4	7	3	0
0	0	0	6	7	12	11	18	14	17	6	4	3
0	2	5	12	10	18	18	26	22	21	13	10	3
3	7	10	19	17	29	30	41	35	33	19	13	3
3	8	11	22	29	38	30	42	37	31	20	8	0
3	9	12	21	23	27	27	32	27	22	13	5	0
6	13	13	16	16	19	21	25	18	11	6	2	0
3	6	6	6	6	8	9	8	6	2	0	0	0

다시하기

14. 제자리 (8점)

각각 1, 2, 3, 4, 5, 6이 적힌 카드가 순서대로 놓여있다. 이때, 왼쪽에서 i번째에 위치하면서 i가 적혀 있는 카드를 "제자리 카드"라고 하자. 정의에 따라 처음에 모든 카드는 "제자리 카드"이다.

당신은 2 이상의 수가 적혀 있는 "제자리 카드" 중 원하는 것을 클릭해서 맨 앞으로 옮길 수 있고, 카드를 옮길 때마다 이동 횟수가 1씩 증가한다. 카드를 적절히 옮겨서 이동 횟수를 최대화하라.



15. 숫자 카드 (11점)

1부터 9까지의 숫자로 구성된 79장의 카드가 일렬로 놓여 있다. 아래에는 지면의 한계상 한 줄에 최대 20개의 카드가 표시되어 있으나, 실제로는 일렬로 놓여 있다고 간주하자. 위치가 i인 카드는 왼쪽에서부터 i번째에 놓여 있는 카드이다.

연속하는 5장의 카드를 한꺼번에 제거할 수 있다. 제거할 카드 중 가장 왼쪽 카드를 클릭하면, 해당 카드부터 시작해서 오른쪽으로 연속한 5장의 카드가 제거된다. 이러한 작업을 15번 반복하여, 4장의 카드를 남겨야 한다.

카드를 적절히 제거하여, 남은 4장의 카드에 적힌 숫자를 이어 붙여 만든 네 자리 수를 최소화하라.

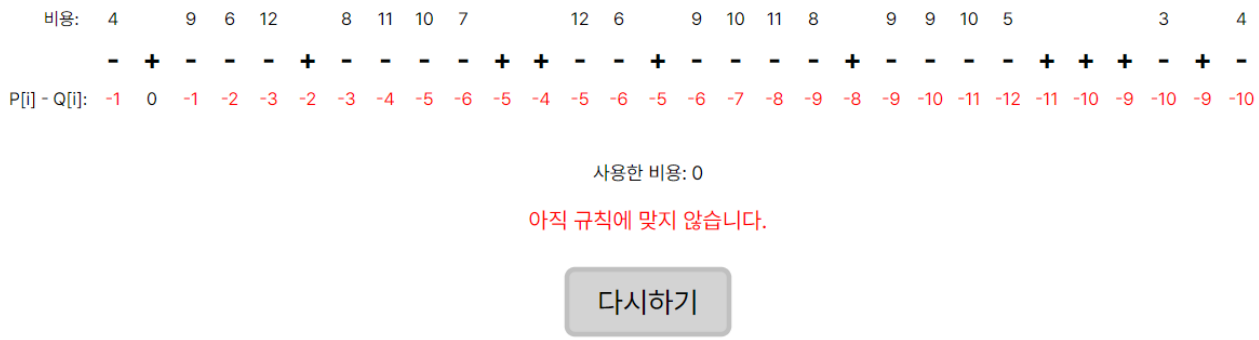
위치:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	7	**4**	**2**	**3**	**1**	**7**	**2**	**7**	**6**	**2**	**8**	**4**	**6**	**4**	**7**	**5**	**8**	**5**	**3**	**5**
위치:	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
	9	**6**	**2**	**3**	**8**	**6**	**5**	**7**	**8**	**6**	**7**	**4**	**9**	**4**	**1**	**6**	**7**	**8**	**2**	**6**
위치:	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
	5	**9**	**8**	**4**	**8**	**7**	**8**	**7**	**8**	**1**	**8**	**4**	**9**	**7**	**8**	**8**	**6**	**9**	**9**	**5**
위치:	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	
	8	**6**	**8**	**7**	**6**	**9**	**6**	**9**	**6**	**6**	**7**	**4**	**8**	**9**	**2**	**6**	**5**	**8**	**7**	

다시하기

16. Plus Minus (13점)

+ 기호와 - 기호로 구성된 길이가 30인 문자열이 있다. - 기호를 클릭하면 + 로 바꿀 수 있고, - 기호 위의 수만큼의 비용이 든다. 총 비용은 - 기호를 + 기호로 바꾸는 데 사용한 모든 비용의 합이다. 최소한의 비용을 들여 아래 규칙을 만족하도록 하여라.

- **규칙**: 문자열의 앞에서부터 k개의 기호 중 +의 개수를 $P[k]$, -의 개수를 $Q[k]$ 라고 하자. 1 이상 30 이하의 모든 정수 i에 대해, $P[i] - Q[i] \geq 0$ 여야 한다.



17. 식별 코드 (13점)

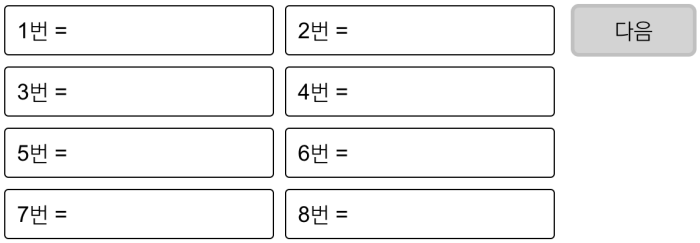
8명의 사람들이 있다. 편의상 각 사람에게 1번 사람에서 8번 사람까지의 번호를 붙이자.

먼저, 당신은 각 사람들에게 식별 코드를 하나씩 부여해야 한다. 식별 코드는 '0'과 '1'로 이루어진 4글자의 문자열이다.

각 사람들에게 식별 코드를 부여하고 [다음] 버튼을 누르면, 개구쟁이 민수가 각 사람들의 식별 코드를 섞은 뒤, 글자를 하나씩 가릴 것이다. 예를 들어, 민수가 식별 코드 '1011'의 두 번째 글자를 가린다면 '1_11'가 남을 것이고, 네 번째 글자를 가린다면 '101_'가 남을 것이다. 당신은 민수가 한 글자씩 가린 각 사람들의 식별 코드를 보고, 원래 각각이 어떤 사람의 식별 코드인지 맞혀야 한다.

당신은 각 사람의 번호와 한 글자씩 가린 식별 코드를 모두 대응시킨 다음 [완료] 버튼을 눌러야 한다. [완료] 버튼을 누르고 나면, 민수가 각 식별 코드가 원래 어떤 사람의 것이었는지를 알려준다. 당신은 식별 코드의 주인을 모두 맞혀야 한다. 즉, 당신이 대응한 식별 코드의 주인과 민수가 알려준 식별 코드의 주인이 모두 일치하도록 식별 코드를 대응해야 한다.

단, 짖궂은 민수는, 당신의 예상과 다르게 식별 코드를 섞는 방법이 존재했다면, 그 방법으로 섞었다고 말할 것이다. 즉, 민수가 식별 코드들에서 어떤 글자를 가려도 주인을 유일하게 대응시킬 수 있도록, 첫 단계에서 식별 코드를 잘 부여했어야 한다.



18. 낚시터 (13점)

격자 위에 낚시터가 있다. 인접한 격자 사이의 거리는 1이다. 당신은 이 낚시터에서 물고기를 최대한 많이 잡으려고 한다.

낚시터에서는 물고기가 특정 시각에 특정 위치에 등장하는 여러 개의 이벤트가 일어난다. 당신은 모든 이벤트에서 물고기가 어떤 시각에 어떤 위치에 등장할지를 알고 있으며, “남은 시간”만큼의 시간이 지난 후 “등장 위치”에 물고기가 등장한다. 물고기가 등장한 순간 당신이 그 위치에 있으면, 당신은 그 물고기를 잡을 수 있다.

당신은 격자를 클릭해 그 위치에서 시각 0에 시작할 수 있으며, 각 시각에 당신은 “왼쪽으로”를 눌러 1의 시간이 지나는 동안 왼쪽으로 거리 1만큼 움직이거나, “기다리기”를 눌러 1의 시간이 지나는 동안 움직이지 않거나, 또는 “오른쪽으로”를 눌러 1의 시간이 지나는 동안 오른쪽으로 거리 1만큼 움직일 수 있다. 단, 1 미만, 20 초과의 위치로는 움직일 수 없다.

물고기를 최대한 많이 잡는 방법을 찾아 제출하여라.

19. 지폐 교환 (16점)

10원, 50원, 100원, 500원, 1000원, 5000원, 10000원, 50000원짜리 지폐가 있다. 지폐는 각각 무작위로 알파벳 A, B, C, D, E, F, G, H, I 중 하나로 표현된다. 각각의 알파벳은 서로 다른 지폐를 가리킨다. 알파벳의 개수가 지폐의 개수보다 하나 많으므로, 남은 알파벳 하나는 어떤 지폐에도 대응되지 않음에 유의하라.

당신은 각각의 지폐가 어떤 알파벳에 대응하는지 알아내고자 한다. 이를 위해, 당신은 다음과 같은 형태의 **질문**을 **정확히 두 번** 할 수 있다.

- X원을 최소 개수의 지폐로 표현하면 어떻게 되는가? (단, X는 10 이상 150 000 이하인 10의 배수)

해당 질문에 대한 **응답**은 아래와 같이 결정된다.

1. X원을 최소 개수의 지폐로 표현한다.
2. 대응되는 지폐의 알파벳들을 알파벳 순서(A, B, C, D, E, F, G, H, I 순서)대로 정렬해서 보여준다.

예를 들어, A가 10원, B가 100원, C가 50원 지폐에 대응될 때, $X = 270$으로 질문했다고 하자. 270원을 최소 개수의 지폐로 표현하면 100원 지폐 2장, 50원 지폐 1장, 10원 지폐 2장이다. 따라서, 이 질문에 대한 응답은 "AABBC"이다.

X 값을 적절히 선택하여 두 번 질문한 후, 응답을 보고 어떤 지폐가 어떤 알파벳에 대응되는지를 맞춰라. 질문을 하는 즉시 응답을 돌려주기 때문에, 첫 번째 질문에 대한 응답을 보고 두 번째 질문을 할 수 있다. 당신이 한 질문과 응답을 통해 지폐에 대응되는 알파벳을 **유일하게 결정**할 수 있어야 정답 처리된다.

"다시하기" 버튼을 누르면 처음부터 다시 시작할 수 있다.

첫번째 X값을 입력하세요: 질의하기 다시하기

아직 완료하지 않았습니다.

20. 사이클의 크기 (16점)

0번부터 $N-1$번까지 N개의 정점으로 이루어진 그래프가 있다. 여기서, N은 $500\,009 \leq N \leq 999\,979$을 만족하는 **소수**이다.

각 $0 \leq i \leq N-2$에 대해, i번 정점과 $i+1$번 정점은 간선으로 직접 연결되어 있다. 또한, $N-1$번 정점과 0번 정점도 간선으로 직접 연결되어 있다. 즉, 그래프는 길이가 N인 사이클이다.

편의상, i번 정점의 왼쪽과 오른쪽 정점을 각각 $(i-1) \mod N$번, $(i+1) \mod N$번 정점이라고 하자. i번 정점에서 왼쪽과 오른쪽으로 k칸 점프한다면, 각각 $(i-k) \mod N$번과 $(i+k) \mod N$번 정점으로 이동하게 될 것이다.

(참고사항) 정수 x에 대해 $x \mod N$은 $x = Nq + r$ (q는 정수, r은 0 이상 $N-1$ 이하의 정수)로 나타냈을 때의 r의 값이다. 예를 들어 $15 \mod 10 = 5$, $-27 \mod 10 = 3$이다.

각 정점은 흰색 혹은 검은색의 색을 가질 수 있다. 처음에 모든 정점은 흰색이다.

어떤 정점 위에 로봇이 놓여 있다.

이제, 여러분은 아래와 같은 형식의 코드를 최대 5번 실행할 수 있다. 코드를 실행하기 위해서 여러분은 코드에서 `( A | B | C )`와 같이 괄호 안에 |로 구분된 곳에 주어진 선지 중 하나를 택하여 코드를 완성해야 한다. 매 실행마다 다른 코드를 완성해도 되며, 여러분은 해당 실행에서 로봇이 몇 번 점프했는지를 즉시 알 수 있다. 단, 모든 실행에 걸쳐 로봇이 점프한 횟수의 총합이 4 096을 초과해서는 안된다.

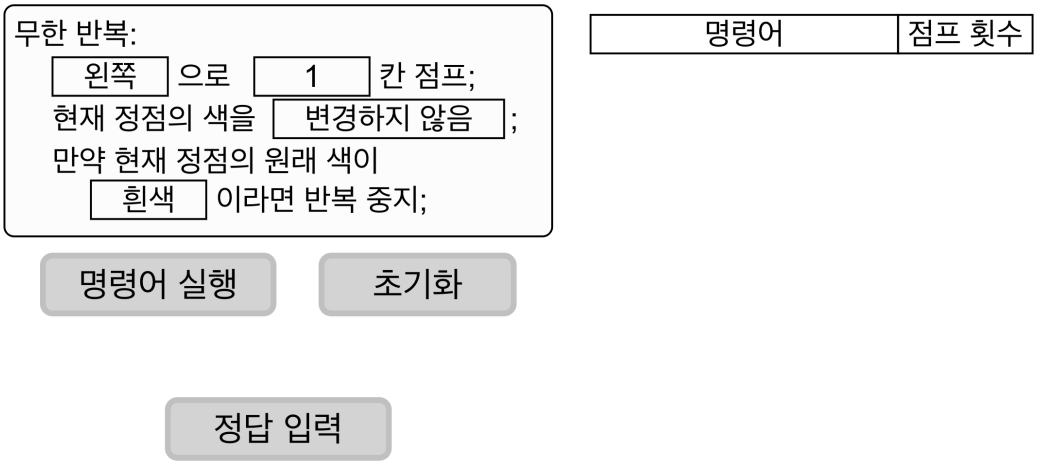
```
무한 반복 {
  로봇이 ( 왼쪽 | 오른쪽 )으로 ( 1 | 10 | 100 | 1,000 )칸 점프;

  현재 정점의 색을 ( 검은색으로 변경 | 흰색으로 변경 | 변경하지 않음 );

  현재 정점의 원래 색이 ( 흰색 | 검은색 )이라면 {
    반복 중지;
  }
}
```

"현재 정점"은 로봇이 현재 놓여 있는 정점을 의미하고, "현재 정점의 원래 색"은 로봇이 현재 정점으로 점프한 직후 해당 정점의 색을 의미한다.

N의 값을 알아내라.

아래에서 '명령어 실행' 버튼을 누르면 코드가 실행된다. N의 값을 알아냈다면, '정답 입력' 버튼을 눌러서 N의 값을 입력하면 된다. 정답을 입력한 이후 "제출" 버튼을 눌러야 한다.

처음부터 다시 시작하고 싶다면, '초기화' 버튼을 누르면 된다. 예를 들어, 로봇이 점프한 횟수의 총합이 4 096을 초과하는 경우, 더 이상 진행할 수 없으므로 초기화 버튼을 눌러 처음부터 다시 시작해야 한다.


무한 반복:
왼쪽
으로
1
칸 점프;
현재 정점의 색을
변경하지 않음
;
만약 현재 정점의 원래 색이
흰색
이라면 반복 중지;
명령어
점프 횟수
명령어 실행
초기화
정답 입력