

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 (정답)
- Ⓜ 350
- Ⓜ 441

2. 덧셈과 뺄셈 (5점)

$4 \square 2 \square 3 \square 5 \square 1 \square 4 \square 3 = 4$ 이다.

괄호 \square 안에는 덧셈 (+) 또는 뺄셈 (-)을 넣어야 한다.

등식을 성립하게 만드는 경우의 수는?

Ⓐ 3

Ⓑ 4

Ⓒ 5

Ⓓ 6 (정답)

Ⓔ 7

3. 10명의 대결 (5점)

어떤 대회에서 한 명의 우승자를 뽑기 위해서 10명의 사람들이 서로 대결하고 있다. 이 대결에서는 2명이 무승부 없이 승패가 결정될 때까지 싸우며, 한 번 대결했던 두 사람이 다시 대결할 수도 있다. 다만, 누구든지 누적 3패를 하면 바로 대회에서 탈락한다. 마지막 한 명만 빼고 다른 사람이 모두 탈락하면 이 사람이 대회의 우승자가 된다.

우승자를 결정할 때 가장 많은 횟수의 대결이 필요한 경우와, 가장 적은 횟수의 대결이 필요한 경우를 생각해보고, 두 경우의 대결 횟수 합을 구하시오.

정답: 56

4. 진법 변환 (7점)

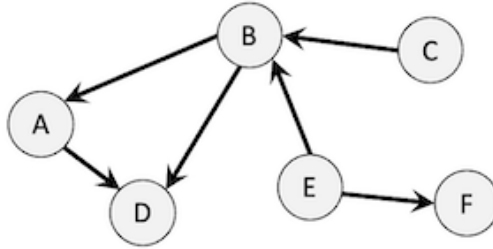
16진수 EA를 8진수로 표현하면 352이며, 이 때 나타나는 각 숫자의 합은 $3 + 5 + 2 = 10$ 이다.

16진수 A6E4F3을 8진수로 표현했을 때 나타나는 각 숫자의 합은?

정답: 31

5. 그래프의 중심 (7점)

방향 그래프 G 에서, 어떤 정점 u 가 **중심 정점**이라는 것은, 정점 u 에서 출발해서 2개 이하의 간선을 거쳐서 다른 모든 정점을 방문할 수 있다는 것이다.



위의 그래프는 중심 정점이 없는 그래프이며, 간선을 몇 개 추가해서 중심 정점이 있도록 만들려고 한다. 최소 몇 개의 간선을 추가해야 하는가?

⊗ 1 (정답)

⊗ 2

⊗ 3

⊗ 4

⊗ 5

6. 순열 찾기 (8점)

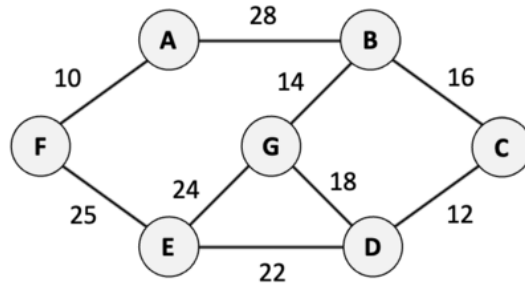
1, 2, 3, 4, 5, 6이 한 번씩 나타나는 여섯 자리 자연수는 총 $6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$ 개 있다.

이 자연수들을 오름차순으로 나열했을 때, 346번째로 나타나는 자연수는 무엇인가?

정답: 362451

7. 최소 신장 트리 (8점)

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



정답: 99

8. 기사, 도둑, 바보 (8점)

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?

정답: 4

9. 인버전의 개수 (10점)

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

c_i 를 $j < i$ 이면서 $a_j > a_i$ 인 j 의 개수라고 하자. c_1, c_2, \dots, 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

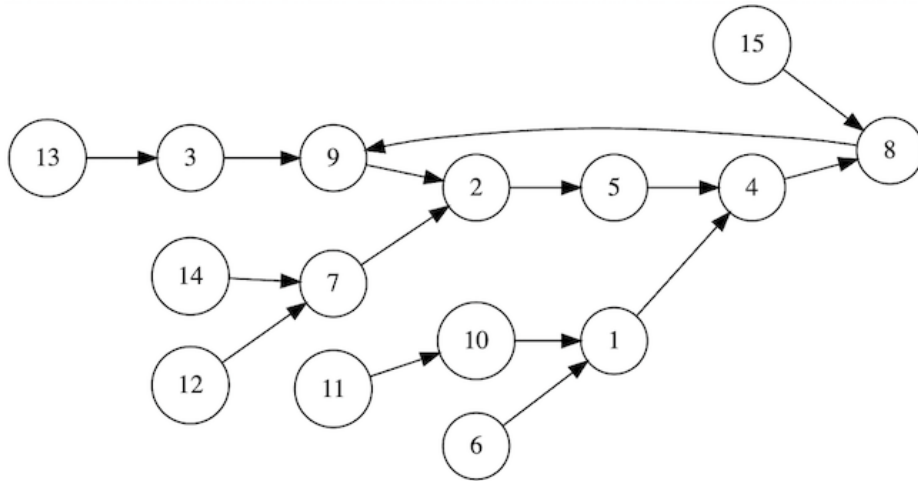
위의 정보를 통해 수열 a 를 유일하게 결정할 수 있다.

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.

정답: 43728165

10. 화살표 따라가기 (10점)

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



$f(u, k)$ 를, u 번 정점에서 시작하여 간선을 k 개 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 v 의 합은?

정답: 55

11. 동전 (14점)

KOI 나라에는 4원, 7원 두 가지 종류의 동전이 있다.

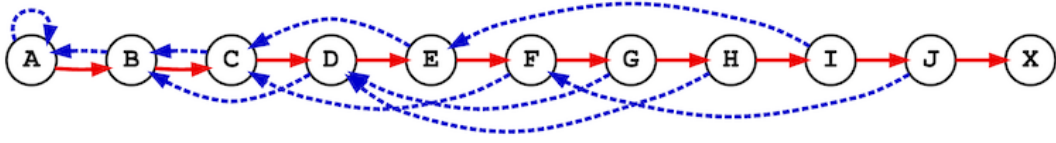
당신은 이 동전으로 물건 값을 내려고 하는데, 이 과정에서 **거스름돈**을 쓸 수 있다. 예를 들어, 10원짜리 물건 값을 내기 위해서 7원 동전 2개를 내고 4원 동전 1개를 거스려 받을 수 있다.

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

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

정답: 37

12. 로봇과 그래프 (14점)



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

빨간색 간선들은 왼쪽에서 오른쪽으로 향하며, 실선 형태 (\rightarrow)로 표시되어 있다. 파란색 간선들은 A A 간선을 제외하고는 오른쪽에서 왼쪽으로 향하며, 점선 형태 (\dashrightarrow)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) \rightarrow B(1) A(3) A(4) \rightarrow B(2) \rightarrow C(1) B(3) ...

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

정답: 1138

13. 과일 (9점)

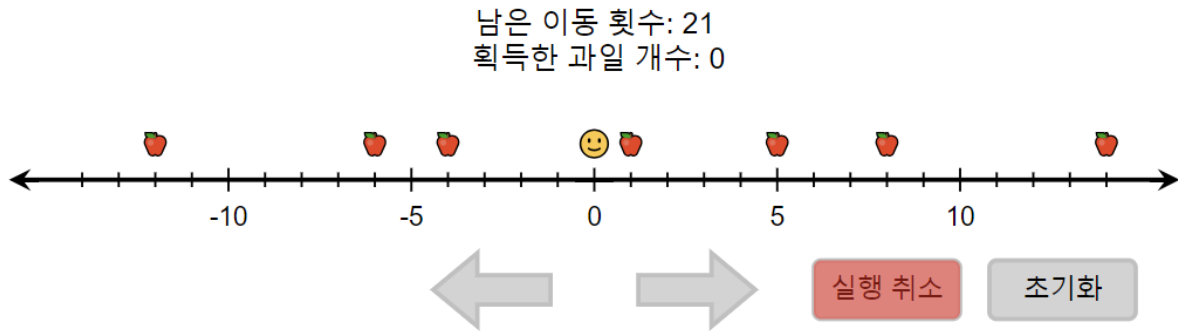
아래와 같이 수직선 위에 과일이 7개 놓여 있다.

여러분은 0에서 출발하여 한 번에 한 칸씩 왼쪽 또는 오른쪽으로 이동할 수 있다. 단, -14보다 작거나 14보다 큰 위치로는 이동할 수 없다.

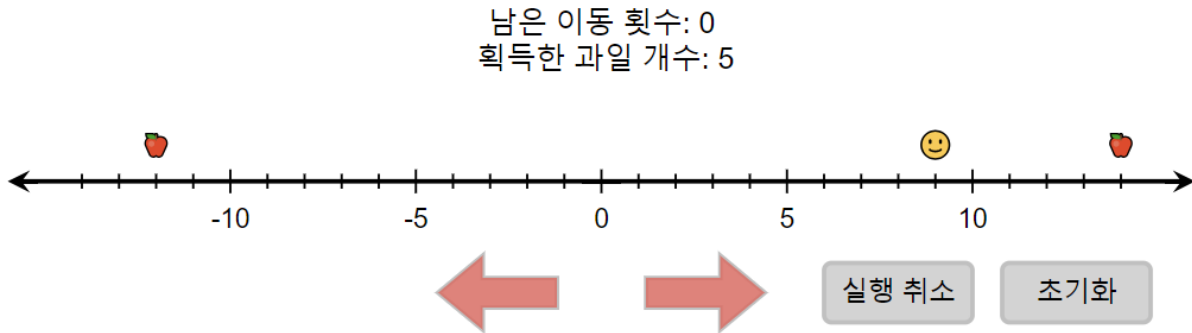
이동하는 도중 여러분의 위치가 어떤 과일의 위치와 같아지면 그 과일을 가져간다.

21회 이내로 움직여서 최대한 많은 과일을 가져가시오.

‘실행 취소’ 버튼을 통해 가장 마지막에 수행한 이동을 되돌릴 수 있다.



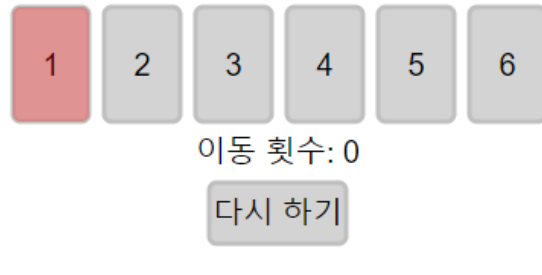
이동 방향을 한 번만 바꾸어도 된다. 왼쪽으로 6번 이동한 후 오른쪽으로 15번 이동하면 아래와 같이 총 5개의 과일을 획득할 수 있다.



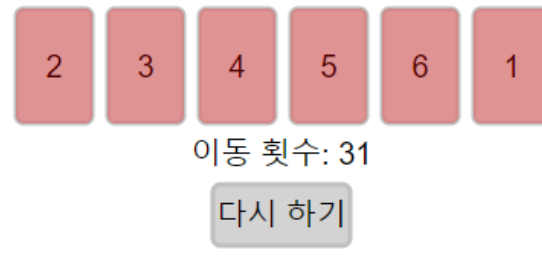
14. 제자리 (9점)

각각 1, 2, 3, 4, 5, 6이 적힌 카드가 순서대로 놓여있다. 이때, 왼쪽에서 i 번째에 위치하면서 i 가 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 최소화하라.



이동할 수 있는 카드 중 가장 앞에 있는 카드를 선택하는 것을 반복하면 된다.



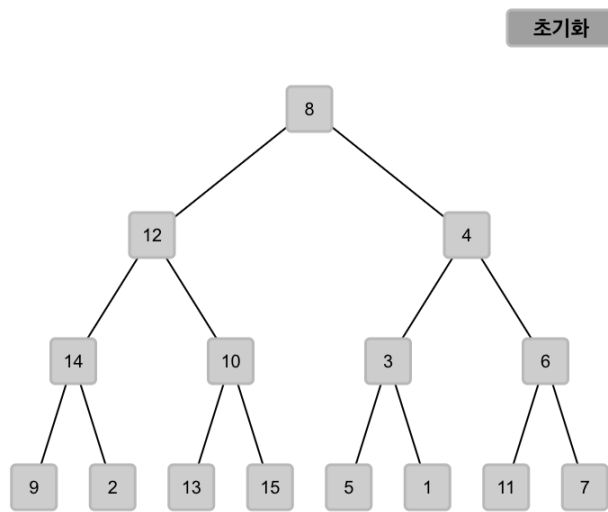
15. 어긋나는 정점 (9점)

아래와 같이 15개의 정점을 가진 이진 트리가 있다.

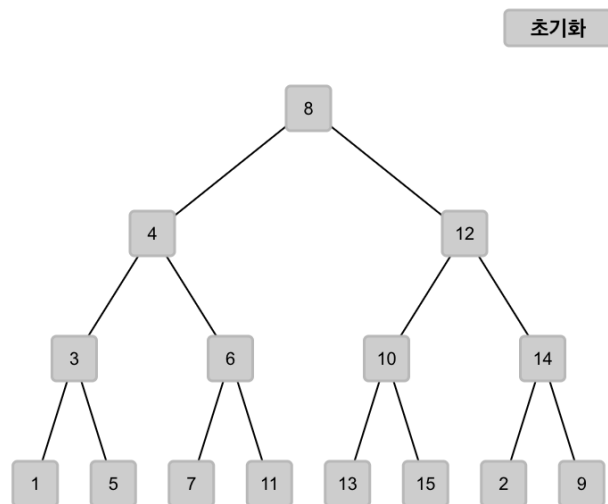
이 트리 상에서 인접한 두 정점 x 와 y 가 다음 조건 중 하나 이상 만족한다면, 두 정점의 쌍 (x, y) 는 **어긋난**다고 하자.

- x 가 y 의 왼쪽 자식이며, x 에 적힌 수가 y 에 적힌 수 보다 크다.
- x 가 y 의 오른쪽 자식이며, x 에 적힌 수가 y 에 적힌 수 보다 작다.

여러분은 말단 정점을 제외한 임의의 정점을 클릭하여 왼쪽 서브트리와 오른쪽 서브트리를 바꿀 수 있다. 어긋나는 인접한 두 정점 쌍의 개수를 최소화하라.



루트 정점에서 시작하여 재귀적으로 탐색하면서, 어긋나는 정점 쌍의 수를 줄이는 방향으로 자식 서브트리를 교환하면 문제를 해결할 수 있다.



16. 숫자 카드 (13점)

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 i 인 카드는 왼쪽에서부터 i 번째에 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

네 자리 수의 $i = 1, 2, 3, 4$ 번째 숫자는 $i \equiv k \pmod{5}$ 를 만족하는 k 번째 카드 중에 하나를 골라야 한다. 네 자리 수를 최소화하기 위해서는 앞 자리부터 최소화하는 것이 이득이므로, 앞 자리부터 최소값을 가지는 카드를 하나씩 고르면 된다.

위치: 1 2 3 4
5 4 7 6

완료했습니다! 반드시 제출 버튼을 눌러주세요.

다시하기

17. Prefix Code (14점)

이진 코드는 0과 1로 이루어진 문자열이다.

아래에 있는 8개의 빈 칸에 이진 코드를 채워야 한다. 이진 코드는 아래 두 조건을 모두 만족해야 한다.

- 빈 칸의 왼쪽에 “길이 l ”이 있는 경우, 해당 빈 칸에 정확히 l 개의 문자를 입력해야 한다.
- 한 이진 코드가 다른 이진 코드의 접두사가 되는 경우가 없어야 한다.

어떤 이진 코드의 접두사는 그 이진 코드를 앞에서부터 읽었을 때 나올 수 있는 모든 이진 코드를 뜻한다. 예를 들어, 101110의 접두사는 1, 10, 101, 1011, 10111, 101110으로 총 6개이다.

동작 설명

- 0 버튼: 코드 맨 뒤에 0을 하나 붙인다.
- 1 버튼: 코드 맨 뒤에 1을 하나 붙인다.
- ← 버튼: 코드 맨 뒤의 문자를 하나 지운다.
- 지우기 버튼: 코드를 전부 지운다.
- 어떤 이진 코드 x 가 다른 이진 코드의 접두사인 경우, x 가 채워진 칸의 글자 색이 연두색으로 변한다.

길이 5	<input type="text"/>	0	1	←	지우기
길이 3	<input type="text"/>	0	1	←	지우기
길이 3	<input type="text"/>	0	1	←	지우기
길이 2	<input type="text"/>	0	1	←	지우기
길이 5	<input type="text"/>	0	1	←	지우기
길이 2	<input type="text"/>	0	1	←	지우기
길이 4	<input type="text"/>	0	1	←	지우기
길이 3	<input type="text"/>	0	1	←	지우기

가장 짧은 코드부터 순서대로 배정하되, 이전 코드들을 접두사로 가지지 않게 배정하면 된다.

길이 5	11110	0	1	←	지우기
길이 3	100	0	1	←	지우기
길이 3	101	0	1	←	지우기
길이 2	00	0	1	←	지우기
길이 5	11111	0	1	←	지우기
길이 2	01	0	1	←	지우기
길이 4	1110	0	1	←	지우기
길이 3	110	0	1	←	지우기

18. Plus Minus (14점)

+ 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 k 개의 기호 중 +의 개수를 $P[k]$, -의 개수를 $Q[k]$ 라고 하자. 1 이상 30 이하의 모든 정수 i 에 대해, $P[i] - Q[i] \geq 0$ 여야 한다.

비용:	4	9	6	12	8	11	10	7		12	6	9	10	11	8	9	9	10	5		3	4								
	-	+	-	-	-	+	-	-	-	+	+	-	-	+	-	-	-	-	+	-	-	-	+	+	+	-	+	-		
$P[i] - Q[i]$:	-1	0	-1	-2	-3	-2	-3	-4	-5	-6	-5	-4	-5	-6	-5	-6	-7	-8	-9	-8	-9	-10	-11	-12	-11	-10	-9	-10	-9	-10

사용한 비용: 0

아직 규칙에 맞지 않습니다.

다시하기

처음으로 $P[i] - Q[i] < 0$ 이 되는 i 에 대해서, 1부터 i 사이에 있는 - 중 최소 비용인 것을 +로 바꿔주는 것을 반복하면 된다.

비용:		9	12		11	10		12		9	10	11		9	9	10	5		3	4										
	+	+	-	+	-	+	+	-	-	+	+	+	-	+	+	-	-	-	-	+	+	+	-	+	+	-	+	-		
$P[i] - Q[i]$:	1	2	1	2	1	2	3	2	1	2	3	4	3	4	5	4	3	2	3	4	3	2	1	0	1	2	3	2	3	2

사용한 비용: 39

완료했습니다! 반드시 제출 버튼을 눌러주세요.

다시하기

19. 식별 코드 (16점)

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 맞춰야 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 맞춰야 한다. 즉, 당신이 대응한 식별 코드의 주인과 민수가 알려준 식별 코드의 주인이 모두 일치하도록 식별 코드를 대응해야 한다.

단, 짓궂은 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.

1번 =	2번 =	다음
3번 =	4번 =	
5번 =	6번 =	
7번 =	8번 =	

이전 모든 식별 코드의 주인을 맞췄습니다! 꼭 제출 버튼을 눌러주세요.

4번 = _110	5번 = _001	1번 = 0000 2번 = 0011 3번 = 0101 4번 = 0110 5번 = 1001 6번 = 1010 7번 = 1100 8번 = 1111
6번 = 1_10	8번 = _111	
3번 = _101	7번 = 1_00	
2번 = 001_	1번 = _000	

그림에 나와 있는 인코딩 외에도 이의 순서를 바꾼 인코딩, 자리별로 0과 1을 바꾼 인코딩 등으로도 문제를 풀 수 있다.

20. 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 질문을 정확히 두 번 할 수 있다.

- X 원을 최소 개수의 지폐로 표현하면 어떻게 되는가? (단, X 는 10 이상 150 000 이하인 10의 배수)

해당 질문에 대한 **응답**은 아래와 같이 결정된다.

1. X 원을 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 X 값을 입력하세요:

질의하기

다시하기

아직 완료하지 않았습니다.

알파벳 i 에 대해서, 튜플 $T[i] = (A[i], B[i])$ 를 생각하자. (단, $A[i], B[i]$ 는 각각 첫 번째, 두 번째 응답에서 알파벳 i 가 등장한 횟수) $T[i]$ 가 모두 다르면 각 알파벳 i 에 대응되는 지폐를 유일하게 결정할 수 있다.

첫번째 X값을 입력하세요:

다시하기

응답 결과: **A A A A B D F F H H H I I I I**

두번째 X값을 입력하세요:

응답 결과: **A A B E G G G G H I I I I**

10원 50원 100원 500원

1000원 5000원 10000원 50000원

지폐 기호:

알파벳을 드래그하여 각각의 지폐에 넣을 수 있습니다.

완료했습니다! 반드시 제출해주세요.