

# 제3회 숙명여자대학교 교내 알고리즘 경진 대회

## SMUPC

Official Solutions



*with*

ALGOS



문제	의도한 난이도	출제자
A Welcome to SMUPC!	Easy	김도은 <sup>숙명여대</sup> <b>whaeun25</b>
B 우당탕탕 영화예매	Easy	김도은 <sup>숙명여대</sup> <b>whaeun25</b>
C 모스 부호	Easy	송혜민 <sup>숙명여대</sup> <b>songfox00</b>
D 이상한 호텔의 송이	Easy	길수민 <sup>숙명여대</sup> <b>2093ab</b>
E 펭귄의 하루	Medium	김도은 <sup>숙명여대</sup> <b>whaeun25</b>
F 벼락치기	Medium	송혜민 <sup>숙명여대</sup> <b>songfox00</b>
G 문자열 만들기	Hard	길수민 <sup>숙명여대</sup> <b>2093ab</b>

# A. Welcome to SMUPC!

출제자 : whaeun25 (김도은, Algos[숙명여자대학교])

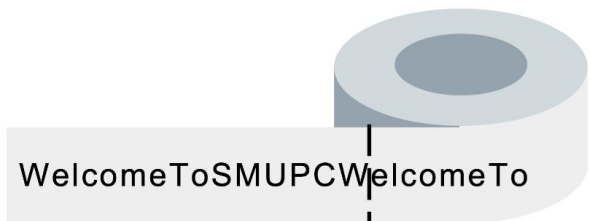
가장 먼저 푼 사람 : 00:02 jungsk20

정답률 : 33.019% ( 35정답 106제출 )

## A. Welcome to SMUPC

### 문제

화은이는 제3회 SMUPC를 맞이하여 환영의 의미로 "WelcomeToSMUPC"가 반복적으로 적혀 있는 라벨지를 프린트했다. 라벨지에는 공백 없이 글자들이 이어져 있고 "WelcomeToSMUPC"의 마지막 글자인 C 이후에는 W부터 다시 "WelcomeToSMUPC"가 반복된다.  $N$ 번째 글자가 있는 곳까지 라벨지를 자르려 할 때,  $N$ 번째에는 어떤 글자가 있을지 구해보자.



[  $N=15$  일 때 라벨지를 자르는 위치 ]

### 입력

첫째 줄에  $N$  ( $1 \leq N \leq 1\,000\,000$ )이 주어진다.

### 출력

첫째 줄에  $N$ 번째에 해당하는 글자를 출력한다.

## A. Welcome to SMUPC

W	e	l	c	o	m	e	T	o	S	M	U	P	C
1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	...

찾고자 하는 문자의 위치  $N$ 을 문자열의 길이인 14로 나눈 나머지를 구한 후 해당 인덱스에 접근해 정해를 구하면 됩니다.

# B. 우당탕탕 영화예매

출제자 : whaeun25 (김도은, Algos[숙명여자대학교])

가장 먼저 푼 사람 : 00:11 jungsk20

정답률 : 33.333% ( 23정답 69제출 )

## B. 우당탕탕 영화예매

### 문제

---

도은이는 동아리 문화의 날을 맞이하여 동아리원들과 함께 좌석이  $M$ 행  $N$ 열의 직사각형 모양으로 배치되어 있는 영화관에서 영화를 보기로 했다. 도은이는 동아리원의 유대감을 중요하게 생각하기 때문에 이미 예매가 완료된 좌석을 피해 동아리원들이 모두 가로로 이어서 앉을 수 있도록 자리를 예매하고 싶어 한다. 도은이를 도와 모든 동아리원들이 가로로 이어서 앉을 수 있도록 예매하는 경우의 수는 총 몇 가지가 있을지 구해보자. 단, 예매한 좌석은 동일하지만, 각 사람이 앉는 위치만 바뀌는 경우는 1가지로 본다.

### 입력

---

첫째 줄에 영화관 세로 줄의 개수  $N(1 \leq N \leq 1\,000)$ 과 가로 줄의 개수  $M(1 \leq M \leq 5\,000)$ , 영화를 관람할 동아리원의 수  $K(1 \leq K \leq 10)$ 가 주어진다.

둘째 줄부터  $N$ 개의 줄에 걸쳐 그 중  $i$ 번째 줄에는,  $i$ 번째 줄의 좌석 예매 상태가 길이  $M$ 인 문자열로 주어진다. 이 문자열은 0또는 1로만 이루어져 있다.  $i$ 번째 줄의  $j$ 번째 문자는  $i$ 번째 줄의  $j$ 번째 좌석의 상태를 나타낸다. 0은 예매할 수 있는 빈 좌석, 1은 예매할 수 없는 예매가 완료된 좌석을 의미한다.

### 출력

---

동아리원들이 모두 가로로 이어서 앉을 수 있도록 영화를 예매하는 경우의 수를 출력한다. 단, 문제에서 주어진 조건에 맞게 영화를 예매할 수 있는 방법이 없다면 0을 출력한다.

- (연달아 가로로 이어져 있는 예매되지 않은 좌석의 개수) - (영화를 관람할 동아리원의 수) + 1의 총 합을 구하면 됩니다.
- 예시로 예매상태로 “0000011000”가 주어지고 영화를 관람할 동아리원의 수가 3명이라면 예매할 수 있는 경우의 수는  $(5 - 3 + 1) + (3 - 3 + 1)$ 인 4가 됩니다.
- $O(NMK)$ 인 브루트포스 풀이도 가능합니다.



# C. 모스 부호

출제자 : songfox00 (송혜민, Algos[숙명여자대학교])

가장 먼저 푼 사람 : 00:20 sksohn01

정답률 : 41.667% ( 25정답 60제출 )

## C. 모스 부호

### 문제

헤민이는 요즘 모스 부호에 관심이 많아졌다. 모스부호는 짧은 신호와 긴 신호를 적절히 조합하여 문자 기호를 표기하는 방식이다. 각 문자를 나타내는 방식은 미리 정해져 있는데, 예를 들어, 짧은 신호는 ' . ', 긴 신호는 ' - '로 나타낸다면, 모스 부호로 알파벳 'A'는 '.-' , 숫자 1은 '-.---'와 같이 표기할 수 있다. 모스 부호를 알고 있으면 위험한 상황에서 구조 요청을 하는 데 유용할 것 같아, 헤민이는 평상시에 친구들과 연락을 주고 받을 때에도 모스 부호를 사용하려고 한다. 헤민이는 친구들이 보내 온 모스 부호를 올바르게 해독했는지 바로바로 확인하고 싶어졌다. 알파벳 A-Z, 숫자 0-9, 기호 ' , ' , ' . ' , ' ? ' , ' : ' , ' - ' , ' @ '로 이루어진 길이  $N$ 인 문자열을 변환한 모스 부호  $N$ 개가 주어질 때, 주어진 모스 부호를 해독하여 원래의 문자열을 출력하는 프로그램을 작성해 보자.

각 문자를 모스 부호로 나타내는 방법은 아래 표에 정리되어 있다. (단, 표의 둘째, 넷째 열은 첫째, 셋째 열의 문자를 모스 부호로 변환한 결과를 나타내며, ' . '는 짧은 신호를, ' - '는 긴 신호를 의미한다.)

### 입력

첫째 줄에 모스 부호로 변환하기 전 문자열의 길이를 나타내는 정수  $N(1 \leq N \leq 100)$ 이 주어진다.

둘째 줄에 원래의 문자열을 모스 부호로 변환한 메시지가 주어진다. 이 메시지에서 짧은 신호는 ' . ', 긴 신호는 ' - '로 나타내며, 원래의 문자열을 구성하는 각각의 문자를 모스 부호로 변환한 결과는 공백으로 구분되어 있다.

위 표를 이용해 해독할 수 없는 메시지는 주어지지 않는다.

### 출력

주어진 모스 부호를 해독하여 길이  $N$ 인 문자열을 공백 없이 출력한다.

알파벳의 경우, 반드시 대문자로 출력한다.

- **map**을 사용해서 풀 수 있는 문제입니다.
- 사용된 알고리즘은 어렵지 않지만 주어진 문자와 모스 부호가 많아서 시간이 걸릴 수 있는 문제입니다.
- **key**에 모스 부호를, **value**에 문자를 저장하고, 주어지는 모스 부호 메시지와 일치하는 **key**를 찾아 **value**값의 문자를 이어붙여 출력하면 됩니다.

# D. 이상한 호텔의 송이

출제자 : 2093ab (길수민, Algos[숙명여자대학교])

가장 먼저 푼 사람 : 01:24 l2x3ge

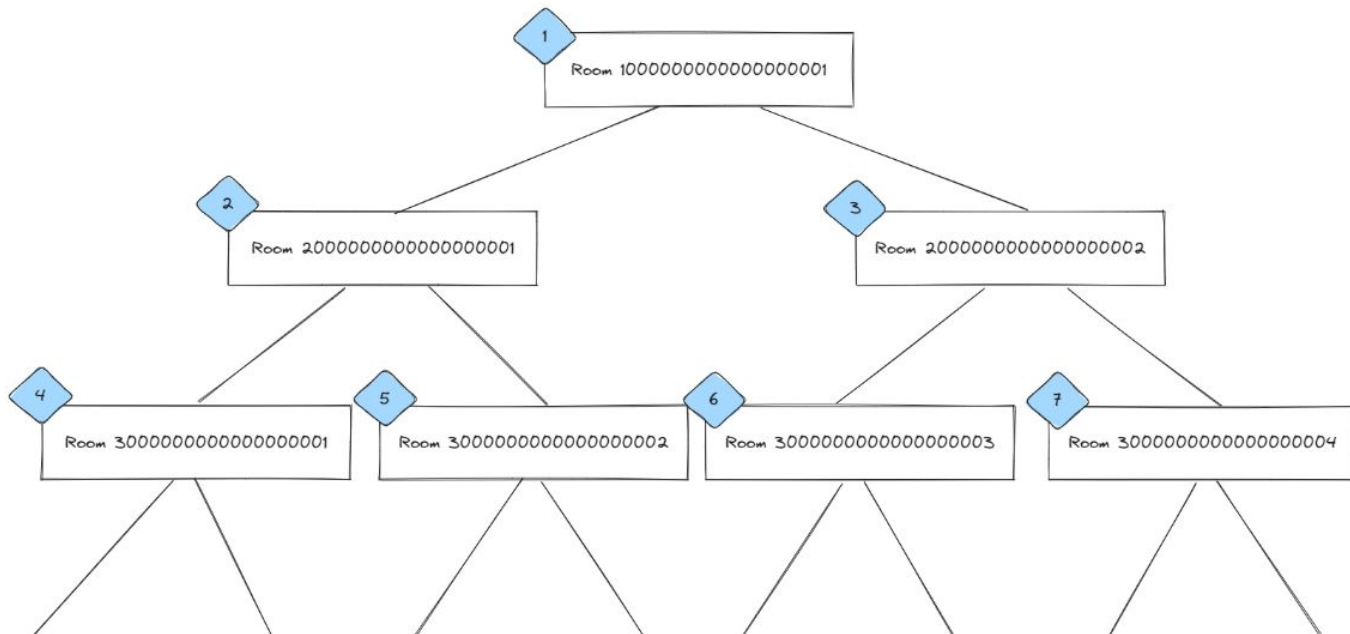
정답률 : 9.524% ( 4정답 42제출 )

#### D. 이상한 호텔의 송이

## 문제

송이가 묵고 있는 S 호텔은 60층짜리 건물인데, 이 호텔의 구조는 완전 이진 트리의 형태로 나타낼 수 있다. 호텔에 있는 방들을 트리의 노드로, 방과 방 사이를 위아래로 잇는 계단을 간선으로 생각한다면, 이 호텔의 구조는 가장 꼭대기에 루트 노드가 있는 완전 이진 트리의 형태이다. 이때, 방과 방 사이를 이동하기 위해서는 두 방 사이를 잇는 계단이 존재해야 하며 계단을 통해서만 이동이 가능하다.

다음 그림은 이 호텔의 1층부터 3층 사이에 존재하는 방과 계단을 나타낸 것이다. 그림에서 직사각형으로 표현된 노드는 방이고 방과 방을 잇는 선은 계단을 의미한다. 직사각형 안에 쓰여있는 수는 그 방의 호수이다. 또, 파란색 사각형 안의 수는 호텔에 있는 모든 방을 호수로 기준으로 오름차순 정렬했을 때 몇 번째로 오는지를 나타내는 수이다.



## D. 이상한 호텔의 송이

이 호텔은 특이하게도 가장 꼭대기 층이 1층이고, 아래로 내려올수록 층의 번호가 1씩 증가하여 가장 아래쪽 층은 60층이다. 완전 이진 트리 구조를 갖는 이 호텔에는, 1층에 1개의 방이, 1층을 제외한 나머지 층에는 층마다 바로 위층에 존재하는 방 개수의 두 배만큼의 방이 존재한다.

이 호텔의 각 층에는 방마다 번호가 붙어 있다. 각 층의 방 번호는 1부터 시작해서 1씩 증가하여, 그 층의 가장 마지막 방의 방 번호는 해당 층에 존재하는 방의 개수와 같다. 또, 1층을 제외한 각 층의  $i$ 번 방은 바로 위층의  $\lceil \frac{i}{2} \rceil$ 번 방과 계단으로 연결되어 있다. 예시로 3층 3번 방은 2층 2번 방과 계단으로 연결되어 있다는 사실을 위의 그림에서 확인할 수 있다.

이 호텔의 각 방의 호수는, 그 방이 위치하는 층의 번호와 해당 층에서 그 방의 번호를 이어 붙인 수이다. 이때, 방 번호가 18자리보다 더 작은 자리의 수라면, 18자리가 되도록 앞에 0을 붙인 뒤 층수와 이어 붙인다. 예를 들면 3층의 가장 마지막 방은 3 000 000 000 000 000 004호이다.

길치인 송이는 지금 이 호텔에 존재하는 모든 방의 호수를 오름차순으로 정렬했을 때  $N$ 번째 방에 있다. 송이는 호텔 전망대가 있는 1층 1번 방으로 최소한의 방만을 거쳐서 이동하려고 한다. 송이가 전망대까지 이동하기 위해 지나야 하는 방들의 호수를 모두 출력하여 송이를 무사히 전망대까지 데려다주자.

### 입력

첫째 줄에 테스트 케이스의 수  $T$ 가 주어진다. ( $1 \leq T \leq 10^4$ )

각 테스트 케이스마다 현재 송이가 있는 방의 위치를 나타내는  $N$ 이 한 줄에 하나씩 주어진다. ( $2 \leq N \leq 10^{18}$ )

송이는 현재 호텔에 존재하는 모든 방의 호수를 오름차순으로 정렬했을 때  $N$ 번째로 오는 방에 있다.

### 출력

각 테스트 케이스마다 현재 송이가 있는 방과 전망대 방을 포함하여 송이가 지나야 하는 방의 호수를 한 줄에 하나씩 차례대로 출력한다.

단, 한 테스트 케이스 내에서 송이가 한 번 지나갔던 방을 다시 지나가서는 안된다.

#### D. 이상한 호텔의 송이

- 완전 이진 트리 구조의 특징을 잘 파악해 봅시다.
- $i$ 층에는 총  $2^{i-1}$ 개의 방이 존재하고 1층부터  $i$ 층까지 총  $2^i - 1$ 개의 방이 존재합니다.
- 호수를 기준으로 정렬했을 때  $n$ 번째 방의 층수와 방 번호를 구해보겠습니다.
- 층수는  $2^k > n$ 인 가장 작은  $k$ 의 값을 구하면 됩니다.
- 방 번호는  $n$ 에서 1층부터  $(k-1)$ 층까지 존재하는 방의 수를 뺀 값과 같습니다.
- 즉, 방 번호는  $n - (2^{k-1} - 1)$ 로 구할 수 있습니다.

#### D. 이상한 호텔의 송이

- 트리의 루트 노드로 이동해야 하기 때문에 루트 노드에 도착할 때까지 부모 노드로 이동하면 최소한의 방안을 거쳐서 이동이 가능합니다.
- 호수를 기준으로 정렬했을 때  $n$ 번째 방에 송이가 있다면 부모 노드는 정렬했을 때  $\lfloor n/2 \rfloor$ 번째 방으로 부모 노드를 구해 이동하면 됩니다. ( $n \neq 1$ )
- 호수를 출력하기 위해 `printf` 출력 형식, `cout.width()`, `cout.fill()` 등을 활용할 수 있습니다.



# E. 펭귄의 하루

출제자 : whaeun25(김도은, Algos[숙명여자대학교])

가장 먼저 푼 사람 : 02:22 l2x3ge

정답률 : 5.882% ( 2정답 34제출 )

## E. 펭귄의 하루

### 문제

$1 \times 1$  크기의 정사각형 칸으로 각각 나누어져 있는  $N \times M$ 의 행렬로 표현되는 펭귄 마을이 있다. 펭귄 마을의 정보는 문자 'S', 'H', 'E', 'D', 'F'로 나타난다. E는 천적이 없어 펭귄이 이동해도 괜찮은 안전 구역을 나타내며, D는 펭귄의 천적인 바다표범이 살고 있어 펭귄이 이동할 수 없는 위험 구역을 나타낸다. 그리고 F는 펭귄이 먹이를 구할 수 있는 물고기 서식지를 의미한다.

펭귄 마을에서 펭귄은 위험 구역이 아닌 곳을 상하좌우로 이동한다. 단, 펭귄은 멸종위기 동물이기 때문에 멸종 위기 동물 보호 구역인 펭귄 마을 밖으로는 이동할 수 없다.

펭귄이 현재 위치에서 출발하여 물고기 서식지 중 최소 한 곳을 둘러 사냥을 마치고 집으로 돌아가려 한다. 펭귄이 사냥을 하는데 걸리는 시간은 고려하지 않으며 출발 지점에서 먼저 물고기들이 서식하는 구역을 둘러지 않았더라도 펭귄이 사는 집을 지나갈 수 있다. 또한, 물고기들이 서식하는 구역을 들른 후에 펭귄이 출발한 지역을 거쳐 펭귄의 집으로 돌아갈 수 있다. 펭귄 마을에서 한 칸을 이동하는 데 1초가 걸린다고 할 때, 물고기를 사냥해 최대한 빠르게 펭귄의 집에 도달하는 데 걸리는 시간을 구해보자.

### 입력

첫째 줄에는 펭귄 마을의 세로 길이  $N(1 \leq N \leq 1000)$ 과 가로 길이  $M(1 \leq M \leq 1000)$ 이 주어진다.

둘째 줄부터  $N$ 개의 줄에 펭귄 마을의 위치 정보를 나타내는 길이  $M$ 의 문자열이 주어진다. 이 문자열은 S, H, E, D, F로 이루어져 있고, 아래와 같은 의미를 가진다.

- S: 펭귄의 현재 위치
- H: 펭귄의 집
- E: 안전 구역
- D: 위험 구역
- F: 물고기 서식지

펭귄의 현재 위치와 펭귄의 집은 공간에 1개만 있으며 물고기 서식지는 공간에 1개 이상 1000개 이하로 존재한다.

### 출력

펭귄이 물고기 서식지를 둘러 집에 도착할 때 걸리는 최소 시간을 출력한다. 만약, 펭귄이 물고기 서식지를 둘러 집에 도착할 수 없다면 -1을 출력한다.

- 2번의 **BFS**를 사용하여 풀 수 있는 문제입니다.
- **BFS**를 이용해 시작점에서 출발하여 각각의 위치에 도달하는데 걸리는 최소 시간과 도착점에서 출발하여 각각의 위치에 도달하는데 걸리는 최소시간을 구합니다.
- 이후, 각 물고기 서식지 위치에 대해 “시작점에서 물고기 서식지가 있는 위치까지의 최소 거리” + “도착점에서 물고기 서식지가 있는 위치까지의 최소거리”가 가장 작은 것을 구하면 됩니다.
- 만약, 펭귄이 물고기 서식지를 들렀다 집에 도착하는 것이 불가능하다면 -1을 출력합니다.

# F. 벼락치기

출제자 : songfox00 (송혜민, Algos[숙명여자대학교])

가장 먼저 푼 사람 : 01:25 sksohn01

정답률 : 5.195% ( 4정답 77제출 )

## F. 벼락치기

### 문제

숙명여자대학교의 알고리즘 학회 ALGOS에 합격한 헤민이는 너무 기빠 마음이 들뜬 나머지 프로그래밍 과제가 있는 것을 잊어버리고 말았다. 프로그래밍 과제로는 다양한 난이도의 문제  $N$ 개가 주어지고, 앞으로  $T$ 일의 제출 기한이 남아있다. 만약 제출 기한 내에 문제를 제출 못 하면, 제출하지 못한 문제마다 정해져 있는 벌금을 내야 한다. 헤민이는 벌금을 내고 싶지 않기 때문에, 내는 벌금의 총금액이 가능한 한 적어지도록 문제를 풀려고 한다.

문제를 해결하는 데 소요되는 일수와 그 문제를 제출 기한 내에 해결하지 못할 경우 내야 하는 벌금이 주어질 때, 헤민이가 내야 하는 벌금의 최소 금액을 구해보자. 제출 기한  $T$ 일이 지났을 때, 제출하지 못한 문제별 벌금의 합이 헤민이가 최종적으로 내야 하는 벌금이다. 단, 헤민이는 아직 프로그래밍에 익숙하지 않아서 한 번에 한 개의 문제만 해결할 수 있다.

	해결하는 데 소요되는 일수	벌금
문제1	2	5000
문제2	1	1000
문제3	1	2000

예를 들어, 프로그래밍 과제로 위와 같이 3개의 문제가 주어졌다고 가정해보자. 제출 기한이 3일 남았다면, 첫째 날에 3번 문제를 해결하고, 둘째 날과 셋째 날에 걸쳐 1번 문제를 해결하면 2번 문제의 벌금인 1000원만 내면 된다.

헤민이가 가능한 한 적은 벌금을 낼 수 있게 도와주자.

### 입력

첫째 줄에 문제의 개수  $N(1 \leq N \leq 1000)$ 과 남은 제출 기한  $T(1 \leq T \leq 1000)$ 가 주어진다.

둘째 줄부터  $N$ 개의 줄에 걸쳐  $i$ 번 문제를 푸는 데 걸리는 일수  $d_i(1 \leq d_i \leq 1000)$ 와 해당 문제의 벌금  $m_i(1 \leq m_i \leq 5000)$ 이 주어진다.

### 출력

최종적으로 내는 벌금이 최소가 되도록 문제를 풀었을 때, 헤민이가 내야 하는 벌금을 출력한다.

만약, 기한 내에 모든 문제를 해결할 수 있다면 0을 출력한다.

- 동적계획법 (Dynamic Programming)을 사용해서 풀 수 있는 문제입니다.
- $dp[i][j]$ 는  $i$ 번째 문제까지 고려했을 때,  $j$ 일 안에 풀 수 있는 문제의 벌금의 합을 나타냅니다.
- 해결한 문제들의 벌금의 합은  $dp[N][T]$ 입니다.
- 따라서 문제별 총 벌금의 합을 저장해뒀다가  $dp[N][T]$ 을 뺀 값이 최종적으로 내는 벌금의 합이 최소가 됩니다.

- $j$ 를 1부터  $T$ 까지 순회하면서  $dp[i][j]$ 를 계산합니다.
- $i$ 번째 문제를 풀지 않았을 경우, 이전 문제까지 고려했을 때의 벌금을 저장합니다.
  - $dp[i][j] = dp[i - 1][j]$
- $i$ 번째 문제를 푸는 경우,  $i$ 번째 문제를 풀지 않았을 때와  $i$ 번째 문제를 푼 경우 중 더 큰 벌금을 선택합니다.
  - if ( $j \geq d[i]$ ) {  
$$dp[i][j] = \max(dp[i][j], dp[i - 1][j - d[i]] + m[i])$$
  
}
- 문제별 총 벌금의 합에서  $dp[N][T]$ 를 뺀 값을 출력합니다.
  - $sum - dp[N][T]$

# G. 문자열 만들기

출제자 : 2093ab (길수민, Algos[숙명여자대학교])

가장 먼저 푼 사람 : 00:00 ???

정답률 : 0.0% ( 0정답 1제출 )



## G. 문자열 만들기

### 문제

알파벳 대문자 'A - Z', 알파벳 소문자 'a - z', 숫자 '0 - 9'를 원소로 가지는 집합  $C$ 의 부분 집합  $P$ 를 정의역으로 하는 함수  $f(c)$ 가 있다.  $f(c)$ 의 값은 1 이상 9 이하의 자연수이다.

1 이상 9 이하인 자연수  $i$ 에 대해 집합  $S_i$ 를 다음과 같이 정의하자. 즉,  $S_i$ 는 집합  $P$ 의 원소 중, 함수  $f$ 에 대한 함숫값이  $i$ 가 되는 문자들의 집합이다.

$$S_i = \{c \in P | f(c) = i\}$$

$S_i$ 에 속하는 문자가 각각 주어진 때, 다음 함수  $g(x)$ 의 값이  $a$  이상  $b$  이하가 되도록 하는 서로 다른  $x$ 의 개수를 구하여 보자.

$$g(x) = \sum_{i=1}^{|x|} f(x_i)$$

단,  $x$ 는  $P$ 에 속한 문자들을 이어 붙여 만든 길이가 1 이상인 문자열이고,  $x_i$ 는  $x$ 의  $i$ 번째 문자이다. 또,  $|x|$ 는 문자열  $x$ 의 길이를 의미한다.

### 입력

첫째 줄에 정수  $a$ 와  $b$ 가 공백을 기준으로 주어진다. ( $1 \leq a \leq b \leq 10^{18}$ ;  $b - a < 5\,000\,000$ )

둘째 줄부터 열 번째 줄까지  $S_i$ 의 모든 문자가  $(i + 1)$ 번째 줄에 공백 없이 주어진다.

집합  $S_i$ 의 원소의 개수는 1 이상 50 이하이다.

주어지는 모든 문자는 알파벳 대소문자, 숫자로 구성되어 있고 각 문자는 최대 한 번씩만 등장한다.

### 출력

서로 다른  $x$ 의 개수를  $1\,000\,000\,007 (= 10^9 + 7)$ 로 나눈 나머지를 출력한다. 이 수는 소수이다.

**핵심 1. 선형 점화식을 다음과 같이 세울 수 있습니다.**

- 동적 계획법(Dynamic Programming)을 활용하여 문제를 해결할 수 있습니다.
- $dp[i]$ 를  $g(x)=i$ 인  $x$ 의 개수라고 정의합니다.
- $S_k$ 의 원소의 개수를  $c_k$ 라고 하면 점화식은 아래와 같습니다.

$$dp[i] = \sum_{k=1}^{\min(i,9)} (c_k \times dp[i - k])$$

**핵심 2. 선형 점화식을 행렬로 표현할 수 있습니다.**

$$\begin{pmatrix} dp[i] \\ dp[i-1] \\ dp[i-2] \\ \vdots \\ dp[i-7] \\ dp[i-8] \end{pmatrix} = \begin{pmatrix} c_1 & c_2 & c_3 & \dots & c_8 & c_9 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & & 1 & 0 \end{pmatrix}^{i-9} \begin{pmatrix} dp[9] \\ dp[8] \\ dp[7] \\ \vdots \\ dp[2] \\ dp[1] \end{pmatrix}$$

- 행렬로 점화식을 표현하면 위와 같습니다.
- 분할 정복을 이용한 거듭제곱을 통해 행렬의 거듭제곱을 빠르게 구할 수 있습니다.
- $dp[a]$ 부터  $dp[a+8]$ 의 값을  $O(9^3 \log a)$ 에 구할 수 있습니다.

**핵심 3. 점화식을 다시 한번 적용하면 a부터 b까지의 값을 구할 수 있습니다.**

- $dp[a]$ 부터  $dp[a+8]$ 까지의 값을 모두 구해놓았습니다.
- 이 값을 활용하여 다시 점화식을 적용하면  $dp[a]$ 부터  $dp[b]$ 까지의 값을  $O(9(b-a))$ 에 구할 수 있습니다.

$$dp[i] = \sum_{k=1}^{\min(i,9)} (c_k \times dp[i - k])$$

※ 이외에도 누적 합을 활용하는 방식도 존재합니다.