

대회 안내

1. 대회 관련

- 각 문제를 해결할 때마다 ({대회 후 경과시간}+{해당 문제 오답 횟수 x20분})의 패널티가 가산됩니다.
- 순위는 문제를 많이 푼 순 > 패널티가 적은 순 > 마지막 정답이 빠른 순으로 정해집니다.
- 문의하기기능을 사용해 문제와 직접적으로 관련된 사항에 대해 질문할 수 있습니다.

2. 제출관련

- 제출시에 문제 번호와 사용 언어를 정확히 선택해주세요. 참가자의 실수에 대해 책임지지 않습니다.
- 제출한 코드는 채점 현황에서 확인할 수 있습니다.

3. 채점 관련

- 표준 입출력 이외의 파일을 읽거나 쓰려고 시도하는 경우 런타임 에러로 처리하여 오답으로 인정합니다.
- 병렬연산이나 스레드를 이용하면 안됩니다. 모든 프로그램은 단일 스레드 프로그램이어야 합니다.
- 시스템에 영향을 주는 기능(종료, sleep 등)을 사용하면 무조건 오답처리 합니다.
- 채점 입력 데이터 모두에 대해 올바른 답을 출력하여야 합니다. 부분점수는 없습니다.
- 예제 데이터는 하나의 예제 일 뿐입니다. 실제 채점은 수 많은 비공개 데이터들로 이루어집니다.
- 각 채점 결과의 의미는 아래와 같습니다. 감독관은 오답의 이유를 절대 알려드리지 않습니다.

채점 결과 메시지	의미
대기중	현재 다른 답안을 채점하는 중
컴파일중	채점을 위해 소스코드를 컴파일 하는 중
채점중	채점 서버에서 해당 답안을 채점하는 중
정답	제출한 답안이 정답으로 인정 된 경우
오답	제출한 답안이 틀린 데이터를 출력한 경우
시간 초과	제출한 프로그램이 제한된 시간내에 종료되지 않은 경우
메모리 초과	제출한 프로그램이 제한된 메모리를 초과하여 사용한 경우
컴파일 에러	제출한 코드에 문제가 있어서 컴파일 되지 않는 경우 <ul style="list-style-type: none"> - 단 경고(Warning)은 무시합니다. - 채점 현황에서 소스코드를 확인하면 컴파일 결과를 알려드립니다.
런타임 에러	제출한 프로그램이 실행도중에 문제를 일으킨 경우 <ul style="list-style-type: none"> - Segmentation Faults - 병렬 연산, 멀티 쓰레딩, 시스템 콜 사용 등 - 파일 입출력 시도

4. C/C++ 관련

- 모든 입출력은 stdin, stdout을 통한 표준 입출력만 사용됩니다.
- GCC 컴파일러를 통해 채점됩니다. 비표준 MS C/C++ 라이브러리를 사용할 경우 컴파일 에러가 발생할 수 있습니다.
- C++의 cin/cout은 scanf/printf에 비하여 속도가 현저히 느립니다. 채점 시간에 반영되므로 후자의 이용을 권장합니다.

5. Java 관련

- 답안의 클래스명은 Main.java이어야 합니다. 또한, 소스코드에 Package 선언문을 절대로 넣지 마세요. 오답 처리됩니다.
- 모든 입출력은 System.in, System.out을 통한 표준 입출력만 사용됩니다.
- Java의 특성상 문자열 처리 시 StringBuilder를 활용하는 편이 속도면에서 매우 유리합니다.
- System.in 입력 시 Scanner보다 BufferedReader와 InputStreamReader를 활용하는 편이 훨씬 속도가 빠릅니다.
- Java는 언어와 구조의 특성상, 표기된 시간 제한이나 메모리 제한을 조금 넘겨도 인정합니다. (인정범위 : +1초, +128MB)

C/C++ 레퍼런스 사이트 : <http://en.cppreference.com/w/>

Python 레퍼런스 사이트 : <http://python.org/>

JAVA 레퍼런스 사이트 : <http://docs.oracle.com/javase/7/docs/>

Problem A

우리는 별을 찍는 소년들

시간 제한 : 1 Second

우리는 우리의 대회를 크게 홍보하기 위해 글씨를 대문짝만하게 뽑기로 했다. 그런데 명색이 대회인데 글씨를 일일이 손으로 적을 순 업지 않지 않겠는가? 따라서 우리는 컴퓨터를 이용해서 글씨를 뽑기로 했다. 그리고 멀리서도 보이게 번쩍번쩍 빛나게 LED를 넣을 것이기 때문에 별 모양으로 크게크게 찍어낼 것이다. 자 이제 우리 모두 함께 대회명을 대문짝만하게 찍어보자.

입력

입력은 주어지지 않는다.

출력

HEPC를 아래와 같이 대문짝만하게 출력한다. 각 알파벳 사이에 한 칸의 공백이 존재한다.

예제 입력

예제 출력

	<pre>* * **** **** ** * * * * * **** **** * * * * * * * * **** * ****</pre>
--	---

Problem B

하노이의 탑

시간 제한 : 1 Second

하노이의 탑(Tower of Hanoi)은 퍼즐의 일종이다. 세 개의 기둥과 이 기둥에 꽂을 수 있는 크기가 다양한 원판들이 있고, 퍼즐을 시작하기 전에는 한 기둥에 원판들이 작은 것이 위에 있도록 순서대로 쌓여있다.

게임의 목적은 다음 두 가지 조건을 만족시키면서, 한 기둥에 꽂힌 원판들을 그 순서 그대로 다른 기둥으로 옮겨서 다시 쌓는 것이다.

1. 한 번에 하나의 원판만 옮길 수 있다.
2. 큰 원판이 작은 원판 위에 있어서는 안 된다.

원판의 개수 n 이 주어질 때, 가장 왼쪽에 있는 기둥에서 가장 오른쪽에 있는 기둥으로 모든 원판을 옮기는 횟수를 출력하는 프로그램을 작성하시오.

입력

첫 번째 줄에 테스트 케이스의 수를 의미하는 자연수 T 가 주어진다. 그 다음에는 T 개의 테스트 케이스가 주어진다. 각 테스트 케이스는 숫자 하나로 이루어져있다. 숫자는 가장 왼쪽에 있는 기둥에 쌓인 원판의 개수 N ($1 \leq N \leq 50$)을 의미한다.

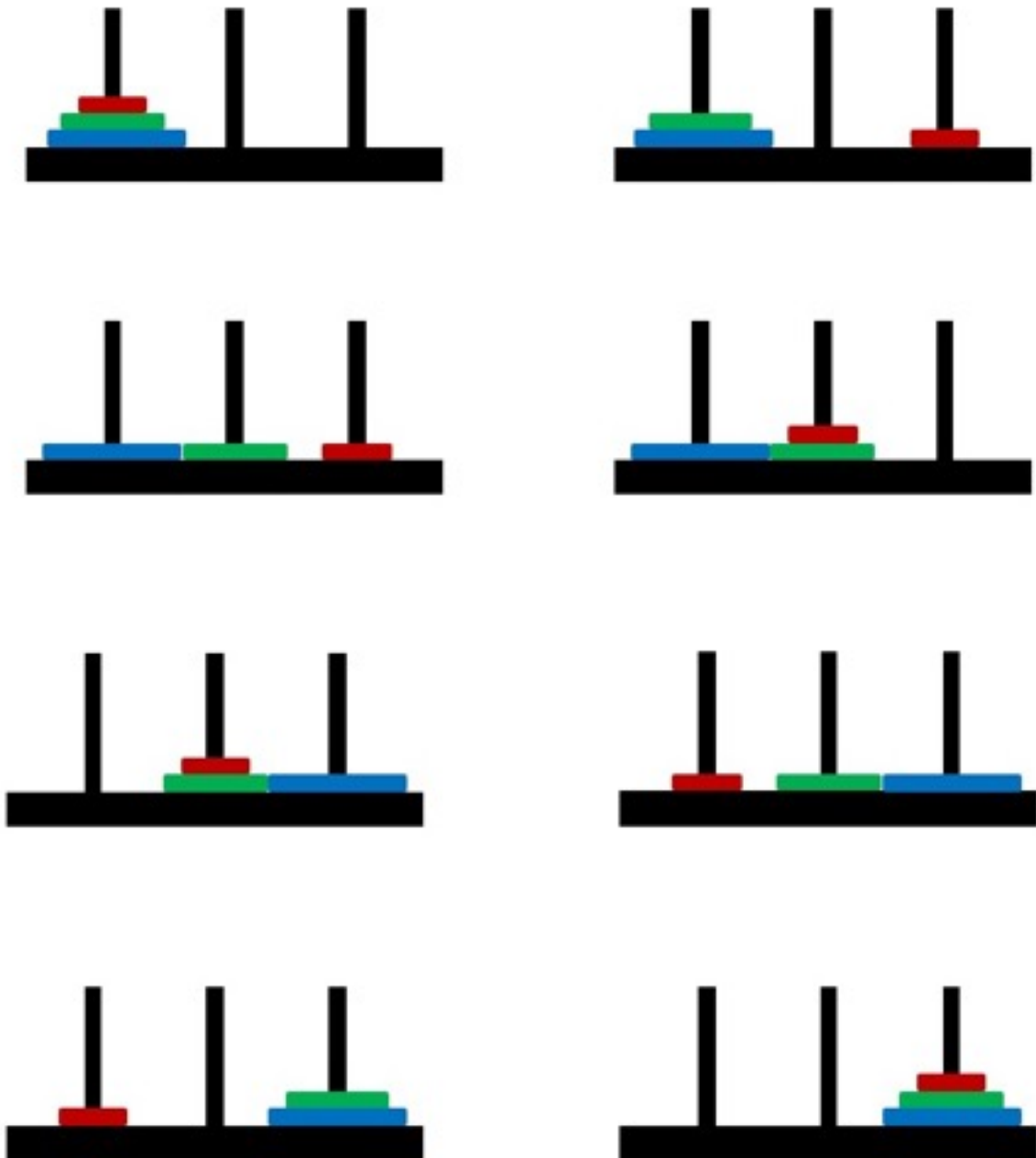
출력

각각의 케이스마다 한 줄에 가장 왼쪽에 있는 기둥에서 가장 오른쪽에 있는 기둥으로 모든 원판을 옮기는 횟수를 출력한다.

예제 입력

2	7
3	31
5	

예제 출력



원판이 3개일때 최소로 움직이는 과정

Problem C

에라토스테네스의 체

시간 제한 : 1 Second

고대 그리스 수학자 에라토스테네스는 소수를 찾는 획기적인 방법을 발견했다. 이 방법은 마치 체로 치듯이 수를 걸러낸다고 하여 '에라토스테네스의 체'라고 부른다. 예를 들어 50까지의 소수를 찾는다고 하면, 우선 1은 소수가 아니므로 제외한다. 그 다음 2는 소수이고, 2를 제외한 나머지 2의 배수는 모두 제외한다. 그 다음 3은 소수이고, 3을 제외한 나머지 3의 배수는 모두 제외한다. 이런 식으로 모든 수를 고려했을 때, 50까지의 소수는 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 49임을 발견할 수 있다. 이 방법을 이용해 N까지의 소수를 모두 출력하는 프로그램을 작성하시오.

입력

첫 번째 줄에 테스트 케이스의 수를 의미하는 자연수 T가 주어진다. 그 다음에는 T개의 테스트 케이스가 주어진다. 각 테스트 케이스에는 숫자 N이 주어진다. ($2 \leq N \leq 1,000,000$)

출력

입력받은 N까지의 소수를 한 줄에 출력한다.

예제 입력

2	2 3 5 7 11 13 17 19
20	2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
50	

예제 출력

Problem D

피보나치킨

시간 제한 : 1 Second

이 세상엔 신의 비율, '황금비'가 있다.

황금비는 n 번째 피보나치 수를 $f(n)$ 이라고 하면, n 이 충분히 클 때, $f(n+1) / f(n)$ 로서 표현할 수 있다. 이러한 황금비는 앵무조개, 달팽이, 해바라기 등에서 관찰되는데, 또한 치킨을 먹을 때에도 관찰된다. 예를 들어 8명에서 먹을 땐 5마리의 치킨, 21명에서 먹을 땐 13마리의 치킨이 필요한 것이다. 또한 이러한 사람 수에 대해 필요한 치킨 수를 '피보나치킨 수'라고 한다.

하지만 7명에서 먹을때는 단순한 피보나치킨 수로 표현 할 수 없다. 그래서 치킨도르프라는 수학자는 이를 피보나치킨 수의 합으로 해결 할 수 있다고 생각했다. 즉, 7명일 경우엔 5명에서 먹을 때의 필요한 치킨수와 2명에서 먹을 때의 필요한 치킨수의 합인 $3 + 1 = 4$ 마리가 되는 것이다.

치킨을 먹는 사람의 수 N 이 주어졌을때, 필요한 치킨의 수를 출력하는 프로그램을 작성하시오.

입력

첫 번째 줄에 테스트 케이스의 수를 의미하는 자연수 T 가 주어진다. 그 다음에는 T 개의 테스트 케이스가 주어진다. 각 테스트 케이스는 숫자 하나로 이루어져있다. 숫자는 치킨을 먹는 사람의 수 N ($1 \leq N \leq 2,000,000,000$)이다.

출력

각각의 케이스마다 한 줄에 필요한 치킨의 수를 출력한다.

예제 입력

2	1
1	4
7	

예제 출력

Problem E

어구전철

시간 제한 : 1 Second

오늘도 프로그래밍 기초 수업에 늦어버린 늦어버린 수형이는 에리카 나라에서 가장 빠른 어구전철을 타고 학교로 가기로 했다. 어구전철을 타고 가는 동안 심심해진 수형이는 어구전철 놀이를 하기로 했다. 어구전철(語句轉綴, anagram) 놀이는 단어나 문장을 구성하고 있는 문자의 순서를 바꾸어 다른 단어나 문장을 만드는 놀이이다. 예를 들어 silent 와 listen 은 문자를 재배치하여 서로를 만들 수 있으므로 어구전철이다. 어구전철은 상당히 빨라 수형이가 어구전철을 모두 찾기 전에 학교에 도착해 버린다. 그래서 수형이는 빠르게 어구전철 놀이의 해답을 찾는 프로그램을 만들어 보기로 했다. 수형이를 도와 어구전철을 찾는 프로그램을 만들도록 하자. 단어의 길이는 4자로만 제한하고, 문자는 0~9자의 숫자로만 구성되어 있다고 가정한다. 어구전철의 모음들을 한 줄씩 오름차순 출력하며, 출력하는 모음의 순서는 모음들 중에 입력순을 따져서 가장 빠른 것 순으로 출력한다.

입력

첫 번째 줄에 테스트 케이스 T가 주어진다. 각 테스트 케이스의 첫 번째 줄에는 탐색할 단어들의 갯수 n이 주어진다. ($0 \leq n \leq 1000$) 두번째 줄에는 n개의 단어가 한 줄로 주어진다.

출력

각 테스트 케이스에 대해 지문의 조건에 맞게 어구전철들을 출력한다.

예제 입력	예제 출력
2	0952 2509
20	1858 8581
0952 5239 1270 8581 7458 3414 7906 2356	5784 7458 8457
4360 3491 6232 5927 2735 2509 5849 8457	1234 1423 2341 4132
9340 1858 8602 5784	
4	
4132 2341 1234 1423	