



전국 대학생 프로그래밍 대회 동아리 연합
여름 대회 2024

Preliminaries

Official Problemset

예선 문제

2024년 7월 13일, 14:00 → 17:00

주최

전국 대학생 프로그래밍 대회 동아리 연합

후원



박승원 (veydpz)

김동현 (kdh9949)

정재현 (Gravekper)

pjshwa

임지환 (raararaara)

kclee2172

이종서 (leejseo)

이상현 (evenharder)

김준겸 (ryute)

문제 목록

문제지에 있는 문제가 총 11문제가 맞는지 확인하시기 바랍니다.

- A** 체육은 수학과목입니다
- B** Two trees, twelve forests
- C** 미어캣
- D** 이진 검색 트리 복원하기
- E** 지금 자면 꿈을 꾸지만
- F** 두 배
- G** 석고 모형 만들기
- H** 만보기 대행 서비스
- I** 만들레
- J** 동전 쌍 뒤집기
- K** 나무 심기

모든 문제의 메모리 제한은 1GB로 동일합니다.

문제 A. 체육은 수학과목입니다

시간 제한 0.5 초 메모리 제한 1024 MB

UCPC 초등학교의 체육 시간이 다가왔습니다. 체육 선생님이 학교에 나오지 못해 체육 시간을 대신 맡게 된 수학 선생님 키파는, 이제는 해져서 축구공처럼 보이지도 않는 공을 던져 주고 축구든 피구든 하라고 한 뒤, 공을 갖고 재밌게 노는 아이들을 운동장 구석에서 지켜보며 자신이 초등학교이던 시절의 체육 시간을 떠올렸습니다.

초등학생의 키파는 몸이 약했습니다. 축구도 몇십 초 이상 뛰는 게 힘들어서, 피구도 공이 몸에 맞으면 다음 날 몸이 너무 아파서, 언제나 운동장 구석에 앉아 재밌게 노는 친구들을 바라만 보았습니다. 그러다 심심해지면 운동장 위에 낙서를 했습니다. 지금 아이들을 바라보고 있자니 문득 그때의 자신이 떠올랐습니다. 이제 생각해 보면 그런 생각의 시간이 있었기에 수학이 재밌어진 것이 아닐까 싶기도 합니다. 어느새 키파는 자신도 모르게 초등학교 때처럼 운동장 위에 나뭇가지로 원을 그리고 있었습니다.

초등학교 운동장에 앉아 있었을 때에는 커다란 원을 그리고 싶었습니다. 어린 키파의 눈에도 원은, 구불구불하지만 질서 있는 듯, 동그랗다는 느낌이 참 마음에 들었나 봅니다. 운동장이 모두 내 것이 된 것처럼 그저 이곳을 뛰어다니고만 싶어졌습니다. 뛰어다니며 원을 그리고 싶었습니다. 하지만 체육 선생님이 아니라 수학 선생님이 된 키파는, 뛰어다니기에는 여전히 몸이 그리 좋지 않았습니다. 운동장이 칠판이라고 생각하며 운동장 비슷한 직사각형을 그리고, 그 안에 꼭 찬 원을 그릴 수밖에 없었습니다.

여러분이 대신 키파의 꿈을 이루어 줍시다. 직사각형 모양의 운동장이 주어지면 운동장 안에서 그릴 수 있는 가장 커다란 원을 그려 줍시다. 그리고 나서 키파에게 원의 반지름의 길이가 얼마인지 알려 줍시다.

입력

첫 줄에 운동장의 한 변의 길이 H 가 **미터(m)** 단위로 주어집니다. ($5 \leq H \leq 1000$)

둘째 줄에 운동장의 다른 한 변의 길이 W 가 **미터(m)** 단위로 주어집니다. ($5 \leq W \leq 1000$)

입력으로 들어오는 모든 수는 정수입니다.

출력

첫 줄에 운동장 안에 그릴 수 있는 가장 큰 원의 반지름의 길이를 **센티미터(cm)** 단위로 출력합니다. 입력 조건 하에서 이 값은 정수임을 보일 수 있으므로, 정수 형태로 출력합니다.

입출력 예시

| 표준 입력(stdin) | 표준 출력(stdout) |
|--------------|---------------|
| 8 10 | 400 |
| 5 13 | 250 |

문제 B. Two trees, twelve forests

시간 제한 1 초 메모리 제한 1024 MB

1번부터 N 번까지 N 개 정점과 M 개 간선으로 이루어진 가중치 있는 무방향 단순 그래프 G 에 대해 숲 점수를 다음과 같이 정의한다:

- $F_1, F_2, F_3, \dots, F_M$ 각각을 1번부터 N 번까지 N 개 정점으로 이루어져 있으며 간선이 없는 그래프라 하자.
- G 의 간선들을 가중치 오름차순으로 e_1, e_2, \dots, e_M 이라 할 때, $i = 1, 2, \dots, M$ 에 대해 순서대로 다음을 시행한다:
 - F_j 에 e_i 를 추가했을 때 사이클이 생기지 않는 최소의 양의 정수 j 를 찾아 F_j 에 e_i 를 추가한다. 여기서 e_i 를 추가한다는 것은 e_i 의 양 끝 정점의 번호가 u_i, v_i 일 때 F_j 에 u_i 번 정점과 v_i 번 정점을 잇는 간선을 추가하는 것을 뜻한다.
- F_i 가 하나 이상의 간선을 가지는 가장 큰 i 를 그래프 G 의 숲 점수라 한다.

당신은 양의 정수 k 에 대해 숲 점수가 정확히 k 이고 정점이 2024개 이하인 그래프 G 를 생성하라는 임무를 받았다.

이 문제가 너무 쉬웠던 당신에게는 다음과 같은 추가적인 조건을 만족하는 G 를 찾는 것이 더 흥미롭게 느껴졌다.

- G 의 정점의 개수를 N 이라 하면, 간선의 개수는 $(2N-2)$ 이다.
- G 의 간선 중 $(N-1)$ 개는 빨간색, 다른 $(N-1)$ 개는 파란색으로 칠해서 빨간색 간선만 남겼을 때 트리가 되고, 파란색 간선만 남겨도 트리가 되도록 할 수 있다.

k 가 주어질 때, 조건을 만족시키는 G 를 구하여 출력해 보자.

입력

첫 줄에 정수 k 가 주어진다. ($2 \leq k \leq 12$)

출력

첫 줄에 그래프 G 의 정점의 개수 N 을 출력한다. ($2 \leq N \leq 2024$)

둘째 줄부터 $(2N-2)$ 개의 줄에 걸쳐 i 번째 줄에 세 정수 a_i, b_i, c_i 를 공백을 사이에 두고 출력한다. ($1 \leq a_i, b_i \leq N$; $a_i \neq b_i$; $1 \leq c_i \leq 10^9$) 이는 a_i 번 정점과 b_i 번 정점을 잇는 가중치 c_i 인 간선이 존재함을 나타낸다.

G 는 다음 조건들을 충족해야 한다.

- 모든 간선의 가중치는 서로 다르다. 즉, c_i 끼리는 서로 다르다.
- 출력한 첫 $(N-1)$ 개의 간선은 트리를 이룬다. 마찬가지로, 그 뒤에 출력한 $(N-1)$ 개 간선도 트리를 이룬다.
- 두 개 이상의 간선으로 직접 연결된 정점 쌍이 존재하지 않는다.
- G 의 숲 점수는 k 이다.

입출력 예시

표준 입력(stdin)

표준 출력(stdout)

3

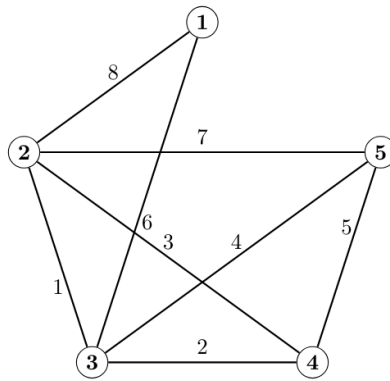
```

5
1 2 8
2 3 1
3 4 2
4 5 5
1 3 6
3 5 4
5 2 7
2 4 3

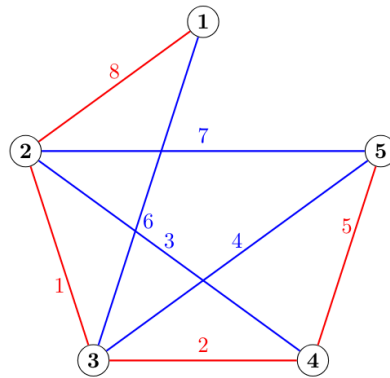
```

노트

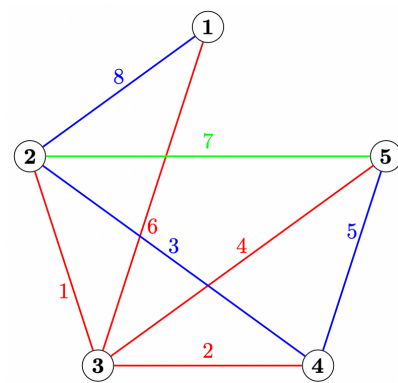
아래는 $k=3$ 인 경우 올바른 답의 예시이다.



위 그래프는 아래 그림에서 확인할 수 있듯 겹치지 않는 두 개의 트리로 구성된다.



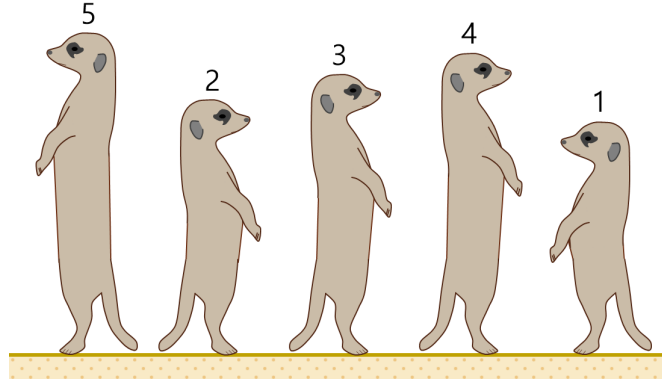
숲 점수를 계산해 보면 아래와 같이 3이 된다. 빨간색 간선은 F_1 , 파란색 간선은 F_2 , 초록색 간선은 F_3 에 소속된 간선을 나타낸다.



문제 C. 미어캣

시간 제한 2 초 메모리 제한 1024 MB

미어캣 N 마리로 구성된 미어캣 가족이 집단생활을 하고 있다. 낮에는 미어캣들이 천적에 대응하기 위해 굴에서 나와 1차원 좌표에서 보초를 선다. 각 미어캣은 보초를 설 때 자신의 위치와 바라보는 방향이 왼쪽 혹은 오른쪽 중 하나로 정해져 있으며, 보초를 서는 동안에는 자신이 바라보는 방향을 바꿀 수 없다.



미어캣 가족 내의 모든 미어캣의 키는 서로 다르며 자신보다 키가 큰 미어캣이 자신이 바라보는 방향에 서 있는 경우 망을 볼 수 없다. 이를 불쌍하게 여긴 당신은 미어캣 가족이 눈치채지 못하게 아래의 행동을 자유롭게 수행할 수 있다.

- 같은 방향을 바라보는 미어캣 둘을 고르고, 서로 자리를 바꾼다.

위의 행동을 적절히 수행했을 때 망을 볼 수 있는 미어캣은 최대 몇 마리인지 구해보자.

입력

첫 줄에 미어캣의 수 N 이 주어진다. ($3 \leq N \leq 5000$)

둘째 줄부터 N 개의 줄에 걸쳐 미어캣에 대한 정보가 주어진다. 모든 $1 \leq i \leq N$ 에 대해, $(i+1)$ 번째 줄에 왼쪽에서 i 번째에 위치한 미어캣의 키를 나타내는 정수 A_i , 바라보고 있는 방향을 나타내는 문자 D_i 가 공백을 사이에 두고 주어진다. ($1 \leq A_i \leq N$) D_i 는 왼쪽이면 L, 오른쪽이면 R로 주어진다.

모든 A_i 는 서로 다르다.

출력

첫 줄에 망을 볼 수 있는 미어캣은 최대 몇 마리인지 출력한다.

입출력 예시

표준 입력(stdin)

표준 출력(stdout)

5
5 L
2 R
3 R
4 R
1 L

4

7
7 R
1 L
6 R
3 L
5 L
4 R
2 R

4

문제 D. 이진 검색 트리 복원하기

시간 제한 2 초 메모리 제한 1024 MB

찬솔이는 N 개의 노드에 정수가 적혀있는 이진 검색 트리를 만들었다. 이진 검색 트리는 모든 노드 i 에 대해서, i 의 왼쪽 자손 노드에 적힌 수는 모두 i 에 적힌 수보다 작고, i 의 오른쪽 자손 노드에 적힌 수는 모두 i 에 적힌 수보다 큰 이진 트리이다.

찬솔이는 모든 i 에 대해서 이진 검색 트리의 i 번 노드의 깊이 H_i 와 이진 검색 트리의 i 번 노드에 적힌 정수 A_i 를 기록해 두었다. 이때, A_i 는 모두 다르고 루트 노드의 깊이는 1이다. ($1 \leq i \leq N$)

찬솔이는 ICPC World Finals에 다녀오는 동안 이진 검색 트리를 잃어버렸다. 찬솔이를 위해 이전에 기록해둔 정보를 이용하여 이진 검색 트리를 복원해 주자.

입력

첫 줄에 노드의 개수 N 이 주어진다. ($1 \leq N \leq 200000$)

둘째 줄부터 N 개의 줄에 걸쳐 트리의 노드 정보가 주어진다. 모든 $1 \leq i \leq N$ 에 대해, $(i+1)$ 번째 줄에는 i 번 노드에 적힌 정수 A_i 와 i 번 노드의 깊이 H_i 가 공백으로 구분되어 주어진다. ($-2 \cdot 10^9 \leq A_i \leq 2 \cdot 10^9$; $1 \leq H_i \leq N$) A_i 는 모두 다르다.

출력

만약 주어진 기록으로 이진 검색 트리를 복원할 수 없다면 -1 을 출력한다.

복원할 수 있다면 트리의 구조를 N 개의 줄에 걸쳐 출력한다.

i 번째 줄에는 i 번 노드의 왼쪽 자식과 오른쪽 자식의 노드 번호를 출력한다. 자식이 없으면 자식의 노드 번호로 -1 을 출력한다.

만약 복원할 수 있는 트리가 여러 개라면 아무거나 복원해도 된다.

입출력 예시

| 표준 입력(stdin) | 표준 출력(stdout) |
|------------------------------|-----------------------|
| 3 1 2 2 1 3 2 | -1 -1 1 3 -1 -1 |
| 3 1 1 2 2 3 2 | -1 |
| 3 2 2 5 3 4 2 | -1 |
| 3 100 2 200 1 300 2 | -1 -1 1 3 -1 -1 |

문제 E. 지금 자면 꿈을 꾸지만

시간 제한 2 초 메모리 제한 1024 MB

“지금 자면 꿈을 꾸지만, 지금 공부하면 꿈을 이룬다.”라는 말이 있습니다. 하지만 실버의 생각은 조금 다릅니다. 실버는 잠을 잘 자서 공부의 효율을 더 높이는 것이 중요하다고 생각합니다. 해야 할 과제가 많았던 실버는 잠을 적당히 자서 최대한 많은 과제를 해내려고 합니다.

실버에게는 N 개의 과제가 있으며, i 번째 과제의 기한은 T_i 입니다. 실버는 시각 0부터 시작하여 원하는 시각에 원하는 과제 하나를 선택하여 진행할 수 있습니다. 과제는 한 번에 하나만 진행할 수 있으며, 과제를 진행하는 도중에 다른 과제를 시작할 수는 없습니다. 실버가 과제 하나를 완료하는 데에는 A 만큼의 시간이 걸립니다.

실버는 0 이상 $(A - 1)$ 이하의 정수 X 를 고른 다음, BX 만큼 잠을 잘 수 있습니다. 이때 잠을 잔 이후에는 과제 하나를 완료하는 데에 $(A - X)$ 만큼의 시간이 걸립니다. 잠은 최대 한 번 잘 수 있으며, 과제를 진행하는 도중에 잠을 잘 수는 없습니다. 또한 시각 0부터 잠을 잘 수도 있습니다.

실버는 잠을 적당히 자서 기한 내에 완료하는 과제의 수를 최대화하려 합니다. i 번째 과제를 정확히 시각 T_i 에 완료한 경우에도 기한 내에 완료한 것으로 생각합니다.

기한 내에 완료할 수 있는 과제의 최대 개수를 구해 봅시다!

입력

첫 줄에 과제의 수 N , 과제 하나를 완료하는데 걸리는 초기 시간 A , 과제 완료 시간 단축의 기준이 되는 정수 B 가 공백을 사이에 두고 주어집니다. ($1 \leq N, A, B \leq 100$)

둘째 줄에 각 과제의 기한을 나타내는 N 개의 정수 T_1, T_2, \dots, T_N 이 공백을 사이에 두고 주어집니다. ($1 \leq T_i \leq 10000$)

입력으로 들어오는 모든 수는 정수입니다.

출력

첫 줄에 기한 내에 완료할 수 있는 과제의 최대 개수를 출력합니다.

입출력 예시

| 표준 입력(stdin) | 표준 출력(stdout) |
|-------------------------------|---------------|
| 3 40 2 70 90 80 | 3 |
| 3 40 10 70 90 80 | 2 |
| 4 30 3 70 75 95 105 | 4 |
| 8 2 5 2 8 9 10 11 12 13 14 | 8 |

문제 F. 두 배

시간 제한 3 초 메모리 제한 1024 MB

밍구는 차세대 채팅 앱 ChatChatA를 개발했다!

ChatChatA를 배포하고 나니, 치명적인 버그 리포트가 밍구에게 날아왔다! 메시지를 입력하는 과정에서, 메시지의 마지막 글자가 특정 글자일 때, 특정 글자를 입력하면 메시지가 “두 배” 된다는 것이다!

정확히 풀어 설명하면 “두 배” 된다는 것은 다음과 같다.

- 현재 입력 중인 메시지를 M , M 의 마지막 글자를 s , 입력할 글자를 c 라고 하자.
- 메시지가 “두 배” 되는 (s, c) 쌍의 집합을 D 라 하자.
 - $(s, c) \in D$ 이면, c 를 입력했을 때 M 은 $M+c$ 가 아닌 $M+M+c$ 가 된다!
 - $(s, c) \notin D$ 이면, c 를 입력했을 때 M 은 $M+c$ 가 된다.
- M 이 빈 문자열이라면, M 은 “두 배” 되지 않는다.

ChatChatA의 사용자인 구밍이는 문자열 T 를 입력하기 위한 최소의 입력 수가 궁금해졌다.

현재 메시지 M 은 빈 문자열로 시작하고, 매 입력은 다음 두 행동 중 하나이다.

- M 에 문자 c 를 추가한다. 위 조건에 따라서 메시지가 “두 배” 될 수 있다.
- M 의 마지막 글자를 삭제한다. 단, M 이 빈 문자열이 아닐 때에만 이 행동을 선택할 수 있다.

현재 메시지 M 이 T 가 되기 위한 최소의 입력 수를 구밍이 대신 구해주자!

입력

첫 줄에 D 의 크기 N 이 주어진다. ($1 \leq N \leq 676 = 26^2$)

둘째 줄에 길이 N 의 영어 소문자 문자열 S , 셋째 줄에 길이 N 의 영어 소문자 문자열 C 가 주어진다.

S 의 i 번째 글자를 S_i , C 의 i 번째 글자를 C_i 라고 하면, $D = \{(S_i, C_i) : 1 \leq i \leq N\}$ 이다.

$|D| = N$ 이다. 즉, 같은 (S_i, C_i) 쌍은 주어지지 않는다.

넷째 줄에 입력할 영어 소문자로 이루어진 문자열 T 가 주어진다. ($1 \leq |T| \leq 500000$)

출력

첫 줄에 현재 메시지를 T 로 만드는 데 필요한 최소의 입력 수를 출력한다.

만약 현재 메시지를 T 로 만들지 못한다면 -1 을 출력한다.

입출력 예시

| 표준 입력(stdin) | 표준 출력(stdout) |
|---------------------------------|---------------|
| 1 t a chatchata | 5 |
| 2 ct ha chatchata | -1 |
| 2 af bd aaafaaafaaafdf | 8 |

문제 G. 석고 모형 만들기

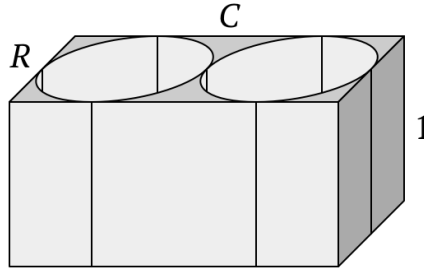
시간 제한 4 초 메모리 제한 1024 MB

현우는 틀에 석고를 붓고 굳혀서 다양한 모양을 만들어 보는 취미가 있다. 현우가 이번에 준비한 틀은 세로 길이가 R , 가로 길이가 C , 높이가 1인 직육면체 모양이다.

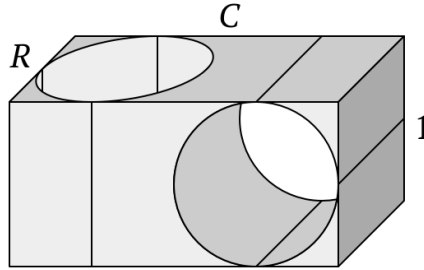
만들어진 모양이 단순하면 석고 모형을 만드는 재미가 떨어진다. 그래서 현우는 지름과 높이가 1인 원기둥을 $R \times C$ 개 가져왔다. 현우는 원기둥을 모두 틀 안에 배치한 다음 빈 공간에 석고를 붓기로 했다.

원기둥을 틀에 배치할 때에는, 틀을 $R \times C$ 개의 단위 정육면체로 나눈 뒤 각 단위 정육면체 안에 꼭 맞게 넣어야 한다. 원기둥을 배치할 수 있는 방향은 세 가지가 있는데, 회전축이 가로를 향하거나, 세로를 향하거나, 바닥에 수직이도록 놓을 수 있다.

현우가 원기둥을 모두 배치하고 나면 틀에 석고를 부어 굳힌 뒤 모든 원기둥을 제거할 것이다. 그러면 여러 개의 분리된 석고 조각이 만들어진다. 예를 들어, 세로 길이가 1이고 가로 길이가 2인 직육면체 틀에 두 원기둥을 회전축이 바닥에 수직이도록 배치한다면 총 6개의 석고 조각이 만들어진다.



한편, 위 예시에서 원기둥 하나의 회전축이 세로를 향하도록 배치를 바꾼다면 총 5개의 석고 조각이 만들어진다.



현우가 틀에 원기둥을 배치하는 방법이 주어졌을 때, 총 몇 개의 석고 조각이 만들어질지 구해 보자.

입력

첫째 줄에 틀의 세로 길이와 가로 길이를 나타내는 정수 R 와 C 가 공백을 사이에 두고 주어진다. ($1 \leq R, C \leq 200$)

다음 R 개의 줄에 걸쳐 현우가 틀에 원기둥을 배치하는 방법이 주어진다. 각 줄에는 길이 C 의 문자열이 주어지며, 문자열을 구성하는 문자의 의미는 아래와 같다.

- **H**: 회전축이 가로를 향하는 원기둥
- **I**: 회전축이 세로를 향하는 원기둥
- **O**: 회전축이 바닥에 수직인 원기둥

출력

첫 줄에 석고를 굳힌 뒤 원기둥을 모두 제거하면 만들어지는 석고 조각의 개수를 출력한다.

입출력 예시

| 표준 입력(stdin) | 표준 출력(stdout) |
|-----------------|---------------|
| 1 2 00 | 6 |
| 1 2 0I | 5 |
| 1 2 0H | 4 |
| 2 2 IH HI | 2 |

문제 H. 만보기 대행 서비스

시간 제한 2 초 메모리 제한 1024 MB

요즘 세상에는 일상생활 속 미션을 수행하면서 포인트를 얻고 소정의 상품도 얻어갈 수 있는 앱테크 서비스가 많다. 만보기 미션은 여러 앱테크 서비스에서 널리 사용하고 있는데, 매일 Dm 걸기에 성공하면 약간의 포인트를 얻을 수 있다.

매일 Dm 씩 걷는 게 생각보다 번거로운 일이기 때문에, 미션을 직접 수행하기 귀찮아하는 사람들을 위해 한별이는 만보기 미션을 대행해주는 사업을 하는 스타트업 스타트한별을 창업하였다. 스타트한별에서는 우선 스타트한별 사옥을 지나는 동서쪽으로 길게 이어진 도로 위에 $1m$ 간격으로 보관함을 설치하고 정수 번호를 매겼다. 스타트한별 사옥에서 동쪽으로 Am 떨어진 보관함의 번호는 A , 서쪽으로 Am 떨어진 보관함의 번호는 $-A$, 사옥에 있는 보관함의 번호는 0 이다.

당신은 스타트한별 사옥에서 출발하여 모든 고객의 미션을 수행한 다음 회사로 복귀해야 한다. 당신이 업무를 시작하기 전에 이미 모든 고객이 X_i 번 보관함에 휴대폰을 놓아두었다. 당신은 X_i 번 보관함까지 가서 휴대폰을 직접 집어야 하고, 집은 이후 Dm 이상 움직인 다음 X_i 번 보관함에 휴대폰을 반납해야 한다. 당신은 충분히 큰 가방을 갖고 업무를 수행하기 때문에 여러 휴대폰을 동시에 넣어서 움직일 수 있다. 당신의 이동기록은 근무 기록으로 반영되기 때문에 반드시 도로 위에서만 움직여야 한다.

모든 고객의 미션을 수행하고 복귀하기 위해 움직여야 할 최소 거리를 구하는 프로그램을 작성하여라.

입력

첫 줄에 고객의 수 N 과 미션을 수행하기 위해 걸어야 할 최소 이동 거리 D 가 공백을 사이에 두고 주어진다. ($1 \leq N \leq 1000000$; $1 \leq D \leq 10^9$)

둘째 줄에 각 고객이 휴대폰을 놓은 보관함의 번호를 나타내는 N 개의 정수 X_i 가 공백을 사이에 두고 주어진다. ($-10^9 \leq X_i \leq 10^9$)

휴대폰의 위치가 서로 겹치거나 스타트한별 사옥과 같은 곳에 있을 수 있다.

모든 입력 값은 정수이다.

출력

첫 줄에 고객 N 명의 미션을 모두 수행하고 복귀하기 위해 필요한 최소 이동 거리를 출력한다.

정답이 정수가 아닌 경우, 정답보다 작거나 같은 가장 큰 정수를 출력한다.

입출력 예시

| 표준 입력(stdin) | 표준 출력(stdout) |
|---------------|---------------|
| 3 5 -8 1 5 | 36 |

노트

아래 방법을 사용하는 게 최적이다.

1. 2번째 고객의 휴대폰을 집는다.
2. 3번째 고객의 휴대폰을 집는다.
3. 스타트한별 사옥 동쪽 7.5m 지점까지 이동한 후 돌아와서 3번째 고객의 휴대폰을 반납한다.
4. 2번째 고객의 휴대폰을 반납한다.
5. 1번째 고객의 휴대폰을 집는다.



Prelimiaries – 예선대회

6. 스타트한별 사육 서쪽 10.5m 지점까지 이동한 후 돌아와서 1번째 고객의 휴대폰을 반납한다.
7. 스타트한별 사육으로 이동한다.

문제 I. 민들레

시간 제한 2 초 메모리 제한 1024 MB

동서로 무한히 길게 뻗은 UCPC 도로에 1의 간격으로 무한히 많은 화분을 놓아 민들레 화단을 만들었다. 모든 화분에는 정수 번호가 매겨져 있는데, 0번 화분을 기준으로 동쪽으로 k 만큼 떨어진 화분의 번호는 k 번, 서쪽으로 k 만큼 떨어진 화분의 번호는 $-k$ 번이다. 각 화분에는 최대 한 송이의 민들레만 자랄 수 있다.

민들레는 바람이 불면 그 방향으로 1만큼 떨어진 화분에 홀씨를 날린다. 심거나 바람에 날려 화분에 떨어진 민들레 씨는 그 화분에 다른 민들레가 없으면 빠르게 자라나 다음 날이면 홀씨를 날릴 수 있게 된다. 각 민들레마다 홀씨는 충분히 많아 다 떨어지는 일은 없다.

시우는 로봇을 이용해 N 일에 걸쳐 민들레 화단을 관찰하려 한다. 로봇은 화단을 관찰하는 동안 매일 하루에 하나씩 명령을 입력받아 수행한다. 로봇이 수행할 수 있는 명령의 종류는 다음과 같다.

- **L**: 서쪽으로 바람을 일으킨다. 모든 정수 x 에 대해, x 번 화분에 민들레가 심겨 있고 $(x-1)$ 번 화분에 민들레가 없다면 다음 날 $(x-1)$ 번 화분에 새 민들레가 자란다.
- **R**: 동쪽으로 바람을 일으킨다. 모든 정수 x 에 대해, x 번 화분에 민들레가 심겨 있고 $(x+1)$ 번 화분에 민들레가 없다면 다음 날 $(x+1)$ 번 화분에 새 민들레가 자란다.
- **C x**: x 번 화분에 민들레 씨를 심는다. x 번 화분에 민들레가 없다면 다음 날 새 민들레가 자란다. ($-10^9 \leq x \leq 10^9$; x 는 정수)
- **Q**: 현재 민들레가 있는 화분의 개수를 기록한다.

화단 관찰을 시작하기 전에는 0번 화분에만 민들레 한 송이가 심겨 있다. 시우가 N 일 동안 로봇에게 요청한 명령의 목록이 주어졌을 때, 명령을 수행하는 프로그램을 작성하시오.

입력

첫 줄에 명령의 수 N 이 주어진다. ($1 \leq N \leq 200000$)

둘째 줄부터 N 개의 줄에 걸쳐 한 줄에 하나씩 명령이 주어진다. 그중 i 번째 명령은 로봇이 i 일에 처리해야 하는 명령을 의미한다. ($1 \leq i \leq N$)

명령 중 **Q**는 하나 이상 있다.

출력

명령으로 **Q**가 주어진 날마다, 그날 민들레가 있는 화분의 개수를 한 줄에 하나씩 시간순으로 출력한다.

입출력 예시

| 표준 입력(stdin) | 표준 출력(stdout) |
|--------------|---------------|
| 13 | 5 |
| L | 6 |
| C 4 | 11 |
| L | 13 |
| Q | |
| C 2 | |
| Q | |
| R | |
| C 7 | |
| R | |
| Q | |
| R | |
| R | |
| Q | |

문제 J. 동전 쌍 뒤집기

시간 제한 2 초 메모리 제한 1024 MB

N 개의 동전이 일렬로 놓여 있다. 각 동전은 앞면이나 뒷면이 보이도록 놓여 있다. 당신은 다음의 조작을 원하는 만큼 반복할 수 있다.

- 서로 이웃하면서 같은 면으로 놓여있는 두 개의 동전을 선택하여 둘 다 뒤집는다.

앞면이 보이는 동전을 뒤집으면 뒷면이, 뒷면이 보이는 동전을 뒤집으면 앞면이 보이게 된다.

각 동전의 상태가 주어질 때 모든 동전을 앞면이 보이도록 놓을 수 있는지 판단하고, 가능하다면 그렇게 하기 위해 필요한 조작의 최소 횟수를 구하시오.

입력

첫 줄에 테스트 케이스의 개수 T 가 주어진다. ($1 \leq T \leq 10000$)

테스트 케이스의 첫 줄에는 동전의 개수 N 이 주어진다. ($1 \leq N \leq 1000000$)

테스트 케이스의 둘째 줄에는 각 동전의 상태를 나타내는 길이 N 의 문자열 S 가 주어진다. 모든 $1 \leq i \leq N$ 에 대해, S 의 i 번째 문자는 i 번째 동전이 앞면인 경우 **H**, 뒷면인 경우 **T**이다.

모든 테스트 케이스에 대한 N 의 총합은 1000000을 넘지 않는다.

출력

각 테스트 케이스에 대해, 주어진 상태에서 모든 동전을 앞면이 보이도록 하기 위해 필요한 조작의 최소 횟수를 한 줄에 출력한다. 불가능하다면 -1 을 대신 출력한다.

입출력 예시

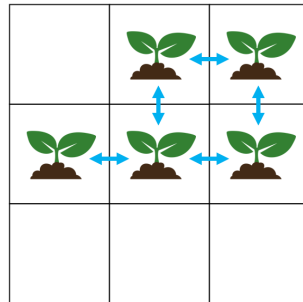
| 표준 입력(stdin) | 표준 출력(stdout) |
|--------------|---------------|
| 2 | 3 |
| 5 | -1 |
| HTHHT | |
| 2 | |
| HT | |

문제 K. 나무 심기

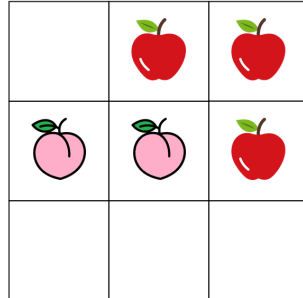
시간 제한 2 초 메모리 제한 1024 MB

승연이는 나무를 심으려 한다. 이를 위해 무한 격자 위에서 몇 개의 칸을 골라 나무 씨앗을 뿌릴 것이다. 나무 씨앗에는 신비한 힘이 있어, 주변 나무 씨앗의 영향을 받아 사과나무 혹은 복숭아나무 중 하나로 자라난다.

씨앗이 자라 어떤 나무가 되는지는 다음 규칙을 따른다. 변을 맞댄 두 칸에 모두 씨앗이 있다면 그 두 씨앗을 서로 **인접**하다고 한다. **인접**한 씨앗의 수가 짝수인 씨앗은 사과나무가 된다. 반면 **인접**한 씨앗의 수가 홀수인 씨앗은 복숭아나무가 된다.

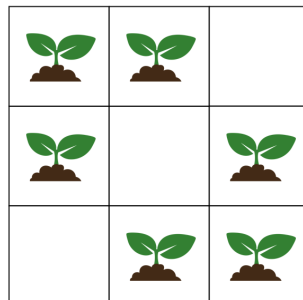


승연이가 위와 같이 5개의 씨앗을 심었다고 하자. 서로 **인접**한 씨앗은 파란색 화살표와 같이 나타난다. 가운데 씨앗은 3개의 씨앗과 **인접**해 있으므로 복숭아나무가 된다.



결과적으로 위와 같이 3개의 사과나무와 2개의 복숭아나무가 열리게 된다.

씨앗을 심을 때는 씨앗들이 격자에서 모두 연결되어 있어야 한다. 즉 임의의 두 씨앗에 대해, 서로 **인접**한 씨앗만 거쳐서 두 씨앗 사이를 이동할 수 있어야 한다. 예를 들어 아래와 같은 배치는 불가능하다.



승연이는 사과나무 A 개, 복숭아나무 B 개를 심으려 한다. 이것이 가능한지 여부와, 가능하다면 나무 배치를 출력하라.

입력

첫 줄에 음이 아닌 정수 A, B 가 공백을 사이에 두고 주어진다. ($A \geq 0; B \geq 0; 1 \leq A + B \leq 200$)

출력

첫 줄에, 배치가 가능하다면 **YES**를, 아니면 **NO**를 출력한다.

배치가 가능할 경우, 둘째 줄에 행과 열의 수 R 와 C 를 공백을 사이에 두고 출력한다. ($1 \leq R, C \leq 200$)

셋째 줄부터 R 개의 줄에 걸쳐, 각 줄에 C 개의 문자를 공백 없이 출력한다. 각 문자는 **0** 혹은 **.** 중 하나이며, 씨앗이 있는 자리에 **0**, 없는 자리에 **.**를 출력한다.

입출력 예시

| 표준 입력(stdin) | 표준 출력(stdout) |
|--------------|--|
| 3 2 | YES 3 3 .00 000 ... |
| 1 0 | YES 1 1 0 |
| 7 0 | YES 3 3 .00 000 00. |
| 8 0 | YES 4 4000 .0.0 .000 |
| 9 0 | NO |
| 0 7 | NO |