

Type

1. [PPASCII XOR cipher](#)
2. [영상올리기](#)
3. [짜맞추기](#)
4. [명절](#)
5. [목걸이 만들기](#)
6. [정기관 3 층](#)

1. PPASCII XOR cipher

유형: 소수 판별, 비트 연산

시간 제한: 1초

메모리 제한: 128MB

문제

비트 연산(Bitwise operation)은 한 개 혹은 두 개의 이진수에 대해 비트 단위로 적용되는 연산이다. 그 예로, XOR 연산은 두 값의 각 자릿수를 비교해, 값이 0으로 같으면 0, 값이 1로 같으면 0, 다르면 1을 계산한다.

```
      0101
XOR   0011
= 0110
```

XOR 연산을 이용하여 암호화 및 복호화가 가능하다.

예를 들어, 문자열 "UOSCS" (01010101 01001111 01010011 01000011 01010011, 8-bit ASCII)는 "AB" (01000001 01000010, 8-bit ASCII)가 반복되는 키(key)로 다음과 같이 암호화가 가능하다.

여기서 "\r"은 carriage return으로 개행을 의미한다.

```
      01010101 01001111 01010011 01000011 01010011 "UOSCS"
XOR   01000001 01000010 01000001 01000010 01000001 "ABABA"
-----
= 00010100 00001101 00010010 00000001 00010010 "\r"
```

다음과 같이 복호화도 가능하다.

```
      00010100 00001101 00010010 00000001 00010010 "\r"
XOR   01000001 01000010 01000001 01000010 01000001 "ABABA"
-----
= 01010101 01001111 01010011 01000011 01010011 "UOSCS"
```

ASCII 코드로 작성된 encrypted.txt가 주어졌을 때, 이를 특정 key로 복호화했을 때 온전한 영어 문장들이 나타나는데, 이것의 2번째 문장을 출력하시오.

encrypted.txt를 프로그램에서 읽을 때 반드시 ASCII로 인코딩을 해야 한다.

입력

(None - 현재 경로에 encrypted.txt 파일이 위치한다.)

출력

첫째 줄에 key 값이 출력되고, 다음 줄에 2번째 문장이 출력된다.

예제

입력: encrypted.txt

출력:

%) + / 5 ; = C G I O S Y a e g k m q

Our students learn basic theories of advanced computing technology.

2. 영상 올리기

유형: 문자열

시간 제한: 1초

메모리 제한: 128MB

문제

재은이는 잘먹는 특기를 살려서 유튜브에 먹방 영상을 올리고 있다. 재은이는 자신의 홈페이지를 개설하여 유튜브에 있는 영상의 링크를 자신의 홈페이지에서 띄우고 싶어 한다. 유튜브 영상의 링크를 복사하던 도중, 재은이는 유튜브 영상의 링크가 다른 3 가지 형식의 문자열이 있다는 것을 알았다. 하지만 3 가지 형식의 문자열 중 1 가지 형식의 문자열만 홈페이지에 영상을 띄울 수 있었다. [다음]은 재은이가 확인한 문자열이고 이 중에서 세 번째 문자열만 홈페이지에 영상이 제대로 올라간다.

[다음]

첫 번째 <https://www.youtube.com/watch?v=8Amy9CxU1n>

두 번째 <https://www.youtu.be/8Amy9CxU1n>

세 번째 <https://www.youtube.com/embed/8Amy9CxU1n>

그래서 재은이는 3 가지 형식의 문자열 중 어떤 걸로 입력하든지 원하는 형식의 문자열로 바꿔주는 프로그램을 찾아보고 있다. 재은이를 도와서 링크의 문자열을 적절하게 바꿔주는 프로그램을 작성하시오.

입력

첫째 줄에 테스트 케이스의 개수 T 가 주어진다. 각 테스트 케이스는 한 줄로 이루어져 있으며, 최대 200 개의 알파벳 대소문자 및 숫자로 이루어진 문자열이 주어진다. 각 테스트 케이스의 문자열 형식은 [다음]의 3 가지이며, 형식에 벗어나는 문자열이 주어진다면 error 메시지를 출력한다.

출력

각 테스트 케이스에 대해서, <https://www.youtube.com/embed/> 로 시작하는 링크로 바꾸어서 출력한다. 문자열 형식에 벗어나는 문자열이 주어진다면 error 메시지를 출력한다.

예제

4

<https://www.youtube.com/watch?v=8Amy9CxU1n>

<https://www.youtu.be/8Amy9CxU1n>

<https://www.naver.com/8Amy9CxU1n>

<https://www.youtube.com/embed/8Amy9CxU1n>

[입력]

<https://www.youtube.com/embed/8Amy9CxU1n>

<https://www.youtube.com/embed/8Amy9CxU1n>

error

<https://www.youtube.com/embed/8Amy9CxU1n>

[출력]

6

<https://www.youtube.com/watch?v=11789a3789>

<https://www.youtube.be/watch?v=11789a3789>

<https://www.youtu.be/11789a3789>

<https://www.youtube.com/11789a3789>

<https://www.youtube.com/watch?u=11789a3789>

<https://www.youtube.com/embed/11789a3789>

[입력]

<https://www.youtube.com/embed/117893789>

error

<https://www.youtube.com/embed/117893789>

error

error

<https://www.youtube.com/embed/117893789>

[출력]

3. 짝 맞추기

유형: 배열, 문자열

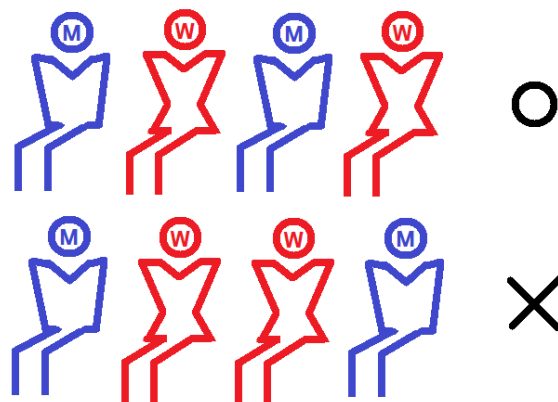
시간 제한: 1초

메모리 제한: 128MB

문제

대학 신입생 주호는 자신의 친구들을 모아 미팅을 시켜주려 한다. 미팅을 하려는 사람들을 한 달 동안 모으고 있던 주호는 예약한 장소에 한 줄로 밖에 앉을 수가 없다는 사실을 뒤늦게 알게 되고 친구들을 위해 한 줄에 앉을 수 있는 미팅 자리배치를 열심히 구상하고 있다.

주호는 자리배치를 위해 미팅을 지원하는 사람들을 왼쪽부터 오른쪽 순서대로 앉히기로 했다. 이때 사람들을 왼쪽부터 순서대로 앉힐 때, 여자(W)는 자신을 기준으로 자신보다 왼쪽에 있으면서 짝을 이루지 않은 남자(M) 중 자신과 가장 가까운 남자(M)와 짝을 이룰 수 있다. 짝을 이룬 남자(M)는 다른 여자(W)와 짝이 될 수 없고, 짝을 이루지 못한 남자(M) 또는 여자(W)가 존재해서는 안 된다.



주호를 도와주기 위해 입력으로 주어진 남녀 문자열이 주호의 생각과 같은 결과인지 판단하여 모두 짝을 이룬다면 yes, 짝을 이루지 못한 사람이 있다면 no 로 나타내주는 프로그램을 작성하시오.

(왼쪽[←], 오른쪽[→]은 스크린을 보는 참가자를 기준으로 함)

입력

입력은 T 개의 테스트 데이터로 주어진다. 입력의 첫 번째 줄에는 입력 데이터의 수를 나타내는 정수 T 가 주어진다. 각 테스트 데이터의 첫째 줄에는 남녀가 앉아있는 자리가 문자열이 한 줄에 주어진다. 남자는 대문자 'M', 여자는 대문자 'W' 로 주어진다. 하나의 문자열의 길이는 2 이상 50 이하이다.

출력

만일 입력 문자열이 주호의 생각과 같은 문자열이면 “yes” 아니면 “no”를 한 줄에 하나씩 차례대로 출력해야 한다.

예제

```
6
MWMW
MWWM
MMWMWWMMMWWW
MMMWMWMMWWWMMMMWWWWMW
MWMWMWMMWMMWMMWWMW
MMWMMMWWMMW
```

[입력]

```
yes
no
yes
no
yes
no
```

[출력]

```
6
MWM
WW
MM
MMW
WM
MW
```

[입력]

```
no
no
no
no
no
yes
```

[출력]

4. 명절

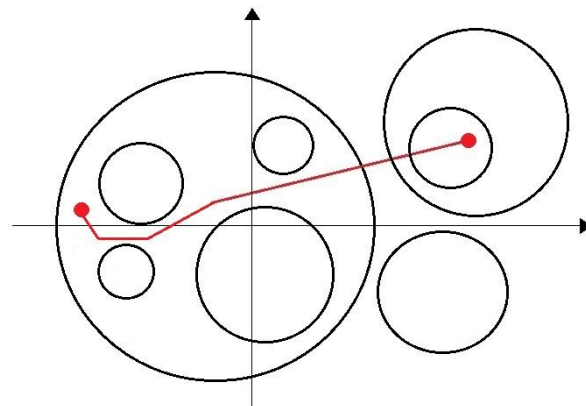
유형: 수학

시간 제한: 1초

메모리 제한: 128MB

문제

명절에 철훈이는 고향을 가기 위해서 자동차를 하나 장만했다. 자동차에 돈을 너무 많이 쓴 철훈이는 고향에 내려갈 때 톨게이트 비를 최대한 아끼려고 생각중이다. 톨게이트 비용은 지역마다 전부 동일하고 지역을 나가거나 들어갈 때 돈을 지불한다. 아래의 그림은 철훈이가 사는 도시의 일부이다.



빨간 실선은 철훈이가 출발점에서 도착점까지 도달하는데 있어서 필요한 지역 진입/이탈 횟수를 최소화 하는 자동차 경로이며, 원은 지역의 경계를 의미한다. 이러한 경로는 여러 개 존재할 수 있지만 적어도 3 번의 지역 진입/이탈이 필요하다는 것을 알 수 있다.

위와 같은 도시 지도, 출발점, 도착점이 주어졌을 때 철훈이에게 필요한 최소의 지역 진입/이탈 횟수를 구하는 프로그램을 작성하시오. (지역의 경계가 맞닿거나 서로 교차하는 경우는 없다고 가정한다. 또한, 출발점이나 도착점이 지역 경계에 걸쳐진 경우 역시 입력으로 주어지지 않는다.)

입력

입력의 첫 줄에는 테스트 케이스의 개수 T 가 주어진다. 그 다음 줄부터 각각의 테스트케이스에 대해 첫째 줄에 출발점 (x_1, y_1) 과 도착점 (x_2, y_2) 이 주어진다. 두 번째 줄에는 지역의 개수 n 이 주어지며, 세 번째 줄부터 n 줄에 걸쳐 지역의 중심과 반지름 (cx, cy, r) 이 주어진다. 입력제한은 다음과 같다. 좌표와 반지름은 모두 정수이다. $(-1000 \leq x_1, y_1, x_2, y_2, cx, cy \leq 1000, 1 \leq r \leq 1000, 1 \leq n \leq 50)$

출력

각 테스트 케이스에 대해 철훈이가 거쳐야 할 최소의 지역 진입/이탈 회수를 출력한다.

예제

```
2
-5 1 12 1
7
1 1 8
-3 -1 1
2 2 2
5 5 1
-4 5 1
12 1 1
12 1 2
-5 1 5 1
1
0 0 2
```

[입력]

```
3
0
```

[출력]

5. 목걸이 만들기

유형: 동적계획법 (Dynamic Programming, DP)

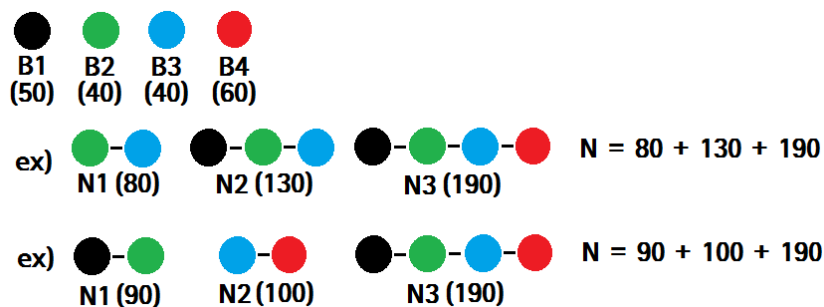
시간 제한: 1초

메모리 제한: 128MB

문제

비즈 공예사인 현서는 자신의 목걸이를 구슬을 이용하여 만든다. 현서는 N 개 구슬에 순서대로 번호를 붙인다. 모든 구슬을 합쳐서 최종적으로 한 개의 목걸이를 만든다. 이 과정에서 두 개의 구슬을 합쳐서 하나의 부분 목걸이를 만들고, 이 부분 목걸이나 구슬을 계속 두 개씩 합쳐서 여러 개의 부분 목걸이를 만들고, 최종적으로는 하나의 목걸이를 만든다. 두 개의 구슬을 합칠 때 필요한 비용이 두 구슬의 가격의 합이라고 가정할 때, 최종적으로 한 개의 목걸이를 완성하는데 필요한 최소비용을 계산하는 프로그램을 작성하시오.

(단, 구슬의 순서를 바꿔서는 안 된다. 또한 처음 구슬과 마지막 구슬은 이어져있는 것이 아니다.)



예를 들어, B1, B2, B3, B4 가 연속적인 네 개의 구슬이고, 구슬 가격이 각각 50, 40, 40, 60 이라고 하자. 이 구슬들을 합치는 과정에서, 먼저 B2 와 B3 를 합쳐서 부분 목걸이 N1 을 만든다. 이 때 비용 80 이 필요하다. 그 다음으로 B1 과 N1 을 합쳐 부분 목걸이 N2 를 만들면 비용 130 이 필요하다. 최종적으로 N2 와 B4 를 합쳐 최종 목걸이를 만들면 비용 190 이 필요하다. 따라서 최종적으로 한 개의 목걸이를 만드는데 필요한 비용의 합은 $80 + 130 + 190 = 400$ 이다. 다른 방법으로 구슬을 합치면 비용을 줄일 수 있다. 먼저 B1 과 B2 를 합쳐 부분 목걸이 N1 을 만들고, B3 와 B4 를 합쳐 부분 목걸이 N2 를 만들고, 최종적으로 N1 과 N2 를 합쳐 목걸이를 만들 수 있다. 이 때 필요한 총비용은 $90 + 100 + 190 = 380$ 이다.

목걸이에 필요한 구슬의 개수와 각각의 가격이 주어졌을 때, 이 구슬들을 하나의 목걸이로 합칠 때 필요한 최소비용을 계산하는 프로그램을 작성하시오.

입력

입력의 첫 줄에는 테스트 케이스의 개수 T 가 주어진다. 각 테스트 케이스의 데이터는 두 개의 행으로 주어지는데, 첫 행에는 구슬의 개수가 양의 정수 N ($3 \leq N \leq 100$) 으로 주어진다. 두 번째 행에는 1 번부터 N 번까지 구슬의 가격을 나타내는 양의 정수 N 개가 주어진다. 구슬의 가격은 10,000 을 초과하지 않는다.

출력

각 테스트 데이터마다 모든 구슬을 한 개의 목걸이로 만드는 필요한 최소비용을 출력한다.

예제

```
2
4
50 40 40 60
10
30 40 50 40 20 50 80 90 10 20
```

[입력]

```
380
1370
```

[출력]

6. 정기관 3층

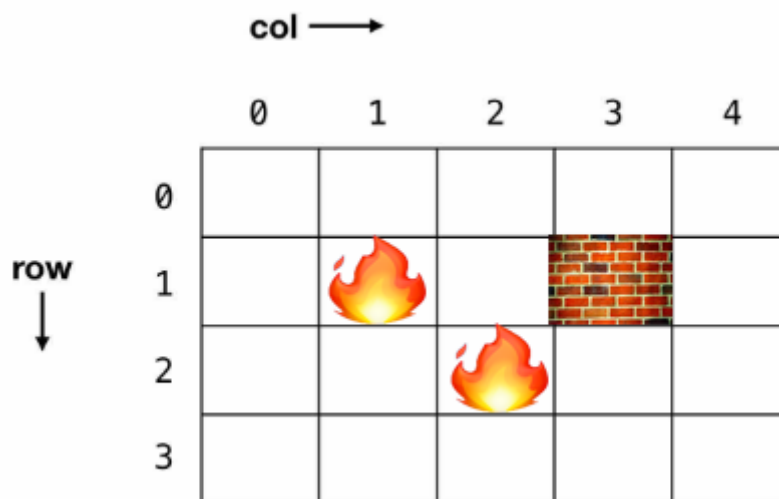
유형: DFS/BFS

시간 제한: 1초

메모리 제한: 128MB

문제

정기관 3층 어느 방에서 불이났다. 매 1 초마다, 불은 동서남북 방향으로 인접한 불이 붙지않은 방으로 퍼져나간다. 벽에는 불이 붙지 않는다.



정기관 3층의 평면도 지도가 주어졌을 때, 얼마나 빨리 정기관 3층의 모든 방이 불에 휩싸이는지 구하는 프로그램을 작성하시오.

입력

첫 줄에는 정기관 3층의 크기를 나타내는 두 정수 r, c 가 주어진다. r 은 정기관 3층의 가로길이, c 는 정기관 3층의 세로 길이를 나타낸다. (단, $2 \leq r, c \leq 100$ 이다.) 둘째 줄부터 r 개의 줄에는 정기관 3층의 방과 벽의 정보가 주어진다. 하나의 줄에는 정기관 3층에 있는 방과 벽의 여부가 c 개의 정수로 주어진다. 정수 1은 불이 붙은 방, 정수 0은 불이 붙지 않은 방, 정수 2는 벽을 나타낸다.

출력

정기관 3층의 모든 방이 불에 휩싸이는 최소 시간을 초단위로 출력한다. 만약, 처음부터 모든 방에 불이 붙은 상태이면 0을 출력해야 하고, 적어도 한 개의 방에 불이 붙지 않는다면 -1을 출력해야 한다.

예제

6 4
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 1

[입력 1]

8

[출력 1]

6 4
0 2 0 0 0 0
2 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 1

[입력 2]

-1

[출력 2]

6 4
1 2 0 0 0 0
0 2 0 0 0 0
0 0 0 0 2 0
0 0 0 0 2 1

[입력 3]

6

[출력 3]

5 5
2 1 0 0 0
0 2 2 2 0
0 2 2 2 0
0 2 2 2 0
0 0 0 0 0

[입력 4]

14

[출력 4]

2 2
1 2
2 1

[입력 5]

0

[출력 5]