

| 문제 1 | |
|--|------------------------------|
| <p>당신은 내일부터 K일 동안 다이어트를 할 계획이다. 한 식당 A에서 점심을, 다른 식당 B에서 저녁을 먹을 것이다.</p> <p>다이어트 중이라 아침은 건너뛰기로 한다. A 식당에는 N 가지의 메뉴가 있고, B 식당에도 N 가지의 메뉴가 있다.</p> <p>당신은 K일 동안 A와 B 식당에서 모두 서로 다른 메뉴를 먹을 것이다.</p> <p>즉, A 식당에서 서로 다른 K개의 메뉴를 고르고 B 식당에서도 서로 다른 K개의 메뉴를 고른 후 이들 메뉴를 적절한 순서로 먹을 것이다.</p> <p>당신이 원하는 것은 K일 중에서 최대 칼로리를 섭취한 날의 칼로리 양을 최소로 하는 것이다.</p> <p>A 식당 메뉴 각각의 칼로리 값과 B 식당 메뉴 각각의 칼로리 값을 입력으로 받아 최대 칼로리를 섭취한 날의 칼로리 양을 최소화하는 프로그램을 작성하라.</p> <p>- 메모리 사용 제한 : heap, global, static 총계 256MB, stack 100MB</p> | |
| <p>입력 파일에는 여러 테스트 케이스가 포함될 수 있다.</p> <p>파일의 첫째 줄에 테스트 케이스의 개수를 나타내는 자연수 T가 주어지고, 이후 차례로 T 개의 테스트 케이스가 주어진다. ($1 \leq T \leq 100$)</p> <p>각 테스트 케이스의 두 식당의 메뉴 수를 나타내는 정수 N과 날짜 수를 나타내는 K가 주어진다 ($1 \leq N \leq 200,000$, $1 \leq K \leq N$).</p> <p>둘째 줄에는 A 식당 메뉴들의 칼로리 값이 양의 정수로 주어진다.</p> <p>셋째 줄에는 B 식당 메뉴들의 칼로리 값이 양의 정수로 주어진다.</p> <p>모든 칼로리 값은 109이하이다.</p> | |
| 입력 | 출력 |
| 2 2 2 1 2 4 2 3 2 6 3 1 1 4 3 | Case #1 5 Case #2 4 |
| 기본 코드 | |
| <pre>/* You should use the standard input/output in order to receive a score properly. Do not use file input and output</pre> | |

Please be very careful.

```
*/
```

```
#include <stdio.h>
```

```
int Answer;
```

```
int main(void)
```

```
{
```

```
    int T, test_case;
```

```
    /*
```

The freopen function below opens input.txt file in read only mode, and afterward,

the program will read from input.txt file instead of standard(keyboard) input.

To test your program, you may save input data in input.txt file, and use freopen function to read from the file when using scanf function.

You may remove the comment symbols(//) in the below statement and use it.

But before submission, you must remove the freopen function or rewrite comment symbols(//).

```
    */
```

```
    // freopen("input.txt", "r", stdin);
```

```
    /*
```

If you remove the statement below, your program's output may not be recorded

when your program is terminated after the time limit.

For safety, please use setbuf(stdout, NULL); statement.

```
    */
```

```
    setbuf(stdout, NULL);
```

```
    scanf("%d", &T);
```

```
    for(test_case = 0; test_case < T; test_case++)
```

```
    {
```

```
////////////////////////////////////
```

```
////////////////////////////////
/*
    Implement your algorithm here.
    The answer to the case will be stored in variable Answer.
*/

////////////////////////////////
////////////////////////////////
////////////////////////////////

    // Print the answer to standard output(screen).

    printf("Case #%d\n", test_case+1);
        printf("%d\n", Answer);

}

return 0;//Your program should return 0 on normal termination.
}
```

문제 2

두 사람 A와 B가 숫자가 적힌 카드들을 가지고 게임을 할 것이다. A와 B사이에는 각각 n 장이 쌓여 있는 카드 더미가 2개 있다.

A부터 게임을 진행할 것이고 둘이 번갈아 가며 한 턴 씩 진행한다.

A와 B는 자신의 턴이 되면 두 더미 중 하나를 골라서 위에서부터 한 장 이상의 카드를 가져가야 한다.

단, 한 턴에 가져가는 카드들에 적힌 숫자의 합이 k 이하라야 한다. 카드들에 적힌 숫자는 모두 k 이하임이 보장된다.

마지막으로 카드를 가지고 가는 사람이 지는 것이 게임의 규칙이다.

두 더미를 X와 Y라고 부르고 두 더미에 주어진 카드들에 적힌 숫자를 아래쪽 카드부터 기록한 것이 각각 X_1, X_2, \dots, X_n 와 Y_1, Y_2, \dots, Y_n 이라고 하자.

우리가 관심이 있는 것은 A와 B가 받을 수 있는 모든 가능한 카드 더미의 상태들을 구별해 보는 것이다.

구별한다의 의미를 설명하기 위해 우선 상태가 무엇인지부터 명확히 하도록 하자.

카드 더미의 한 상태란 X에 i 장이 있고 Y에 j 장이 있는 ($0 \leq i, j \leq n$) 모든 경우들을 각각 말한다.

즉 모든 가능한 상태는 $(n+1)^2$ 가지이다. 참고로, X에 i 장이 있고 Y에 j 장이 있는 상태에서 X에 있는 카드들에 적힌 숫자는

아래쪽 카드부터 X_1, X_2, \dots, X_i , Y에 있는 카드들에 적힌 숫자는 아래쪽 카드부터 Y_1, Y_2, \dots, Y_j 일 것이다.

이 상태를 $\{(X_1, X_2, \dots, X_i), (Y_1, Y_2, \dots, Y_j)\}$ 로 표기하기로 한다.

내 턴이 되었는데 X에 i 장이 있고 Y에 j 장이 있는 상태라면,

1. 내가 최선의 작전을 편다면 상대방이 어떤 작전을 쓰더라도 반드시 이길 수 있는 상황,

2. 내가 지금부터 어떤 작전을 쓰더라도 상대방이 최선의 작전을 쓰면 내가 절대로 이길 수 없는 상황,

두가지의 상황을 생각할 수 있다.

우리는 위의 두가지 종류의 상황에만 관심이 있다.

특별히 $i=0, j=0$ 인 상태는 이미 이긴 것이므로 반드시 이기는 상황으로 간주한다.

예를 들어, $n=2, k=3$ 이라고 하자. 두 더미의 카드에 있는 숫자들이 처음에 $\{(3,3), (3,3)\}$ 인 경우를 생각해 보자.

$k=3$ 이므로 A든 B든 매번 단 한장의 카드만 가져가는 것이 가능하다.

따라서, 상태 $\{(3,3), (3,3)\}$ 를 받으면 반드시 이기고, 상태 $\{(3,3), (3)\}$ 나 $\{(3), (3,3)\}$ 를 받으면 반드시 지고,

상태 $\{(3,3), ()\}$, $\{(), (3,3)\}$ 혹은 $\{(3), (3)\}$ 를 받으면 반드시 이기며,

상태 $\{(), (3)\}$ 혹은 $\{(3), ()\}$ 를 받으면 반드시 진다는 것을 알 수 있다.

| <p>비슷한 예로, $n=2, k=3$이라고 하자.</p> <p>두 더미의 카드에 있는 숫자들이 처음에 $\{(1,2), (1,2)\}$인 경우를 생각해 보자.</p> <p>A는 한 더미에서 한 장을 가져가거나 두 장을 가져갈 수 있다.</p> <p>B가 받은 상태가 $\{(1,2), (1)\}$라면 B는 두 장을 가져가서 $\{(), (1)\}$인 상태를 A에게 주어 반드시 이길 수 있다.</p> <p>또, B가 받은 상태가 $\{(1,2), ()\}$라면 B는 한 장을 가져가서 $\{(1), ()\}$인 상태를 A에게 주어 반드시 이길 수 있다.</p> <p>즉, A가 받은 $\{(1,2), (1,2)\}$인 상태는 B가 최선을 다한다면 A가 어떤 작전을 쓰더라도 이길 수 없는 상황이다.</p> <p>두 카드 더미에 있는 카드들에 적힌 숫자들을 입력으로 받아서, 만들어질 수 있는 모든 가능한 상태들에 대해서 위의 1번과 2번으로 구분되는 상태들의 개수를 계산하는 프로그램을 작성하시오.</p> | |
|--|----------------------------------|
| <p>입력 파일에는 여러 테스트 케이스가 포함될 수 있다.</p> <p>파일의 첫째 줄에 테스트 케이스의 개수를 나타내는 자연수 T가 주어지고, 이후 차례로 T개의 테스트 케이스가 주어진다. ($1 \leq T \leq 50$)</p> <p>각 테스트 케이스의 첫 줄에는 한 더미의 카드 수를 나타내는 정수 n과 한꺼번에 가져갈 수 있는 카드들의 숫자의 합 제한 k가 주어진다 ($1 \leq n, k \leq 3,000$).</p> <p>둘째 줄에는 X더미에 있는 카드들에 적힌 숫자를 아래에서부터 기록한 값들이 양의 정수로 주어진다.</p> <p>셋째 줄에는 Y더미에 있는 카드들에 적힌 숫자를 아래에서부터 기록한 값들이 양의 정수로 주어진다.</p> <p>카드에 적힌 숫자들은 k이하임이 보장된다.</p> | |
| 입력 | 출력 |
| 2 2 3 3 3 3 3 2 3 1 2 1 2 | Case #1 5 4 Case #2 6 3 |
| <p>기본 코드</p> <pre> /* You should use the standard input/output in order to receive a score properly. Do not use file input and output Please be very careful. </pre> | |

```

*/

#include <stdio.h>

int Answer;

int main(void)
{
    int T, test_case;
    /*
        The freopen function below opens input.txt file in read only mode,
and afterward,
        the program will read from input.txt file instead of
standard(keyboard) input.
        To test your program, you may save input data in input.txt file,
        and use freopen function to read from the file when using scanf
function.
        You may remove the comment symbols(//) in the below statement
and use it.
        But before submission, you must remove the freopen function or
rewrite comment symbols(//).
    */
    // freopen("input.txt", "r", stdin);

    /*
        If you remove the statement below, your program's output may not
be recorded
        when your program is terminated after the time limit.
        For safety, please use setbuf(stdout, NULL); statement.
    */
    setbuf(stdout, NULL);

    scanf("%d", &T);
    for(test_case = 0; test_case < T; test_case++)
    {

////////////////////////////////////
////////////////////////////////////
        /*

```

```

        Implement your algorithm here.
        The answer to the case will be stored in variable Answer.
        */

////////////////////////////////////
////////////////////////////////////

        // Print the answer to standard output(screen).

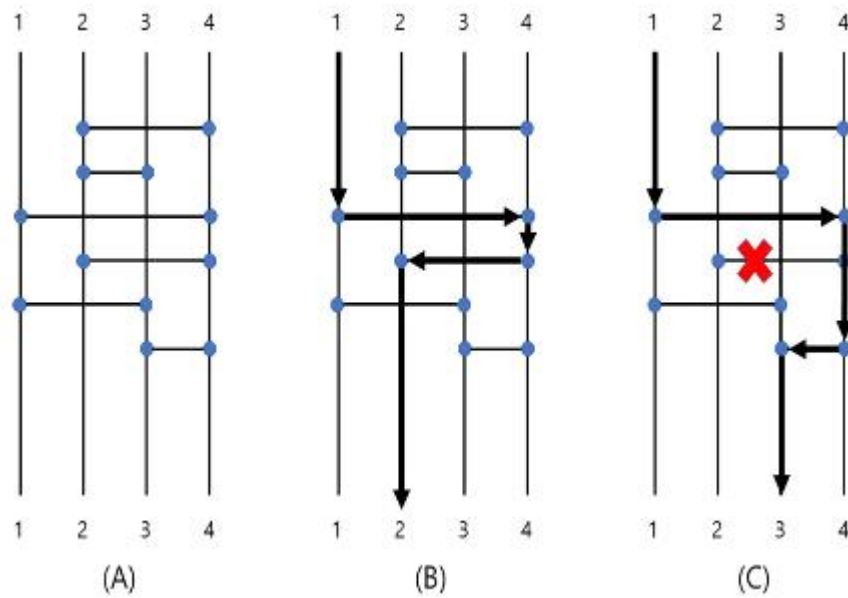
        printf("Case #%d\n", test_case+1);
        printf("%d\n", Answer);

    }

    return 0;//Your program should return 0 on normal termination.
}
```

문제 3

아래 그림 (A)에서 보인 것 같은 사다리가 주어질 때,
 모든 가로방향의 이음선이 정상적인 게임에선 그림 (B)에서 보인 것처럼 상단 1번을 출발
 한 선은 하단 2번에 도착하게 된다.
 즉, 1→2로 연결된다.
 그러나, 그림 (C)에서 보인 것처럼 일부 가로이음선이 고장(X로 표시된 것)이라면 그것을
 통해 이동할 수 없어 1→3으로 연결된다.



사다리에는 N 개의 세로선이 있는데, 그림에서 보인 것처럼 각 세로선은 1부터 N 사이의 정
 수로 구분된다.

k 개의 가로이음선 각각은 양끝점을 연결하는 세로선의 번호 쌍 (i, j) ($1 \leq i < j \leq N$)으로 표현
 된다.

가로이음선의 높이(위치)는 모두 다르며, 위에서부터 아래로 순서대로 그 정보가 주어진다.

출발지점을 나타내는 정수 i ($1 \leq i \leq N$)와 도착지점을 나타내는 정수 j ($1 \leq j \leq N$)가 주어질
 때, i 에서 j 로 가는 경로가 형성되기 위해 가로이음선 중 최소 몇 개가 고장인지를 밝히
 는 프로그램을 작성하시오.

입력 파일에는 여러 테스트 케이스가 포함될 수 있다.

파일의 첫째 줄에 테스트 케이스의 개수를 나타내는 자연수 T 가 주어지고,

이후 차례로 T 개의 테스트 케이스가 주어진다. ($1 \leq T \leq 70$)

각 테스트 케이스의 첫 줄에는 세 정수 N, k, m 이 주어진다.

여기서 N ($2 \leq N \leq 1500$)은 사다리의 세로선의 개수, k ($1 \leq k \leq 2000$)는 가로이음선의 개수,
 m ($1 \leq m \leq 105$)은 출발점과 도착점 쌍의 개수를 나타낸다.

다음 줄부터 k 줄에 걸쳐서 가로이음선의 양끝점을 연결하는 세로선의 번호 쌍 (i, j) ($1 \leq i <$

다음 줄부터 m 개의 줄에 걸쳐서 출발점과 도착점의 위치를 나타내는 정수 쌍 (i, j) ($1 \leq i, j \leq N$)이 한 줄에 한 쌍씩 주어진다.

| 입력 | 출력 |
|---|---|
| 3 3 4 4 1 2 2 3 1 2 1 3 1 1 1 2 2 3 3 2 4 6 5 1 2 2 3 3 4 1 3 2 4 1 4 1 3 1 4 3 3 3 4 4 4 4 2 4 1 2 2 3 1 1 1 2 1 3 1 4 | Case #1 3 Case #2 6 Case #3 1 ● 만약 출발점에서 도착점으로 연결되는 방법이 없을 경우 -1을 더하면 된다. |
| 기본 코드 | |
| <pre>/* You should use the standard input/output in order to receive a score properly.</pre> | |

Do not use file input and output

Please be very careful.

```
*/
```

```
#include <stdio.h>
```

```
int Answer;
```

```
int main(void)
```

```
{
```

```
    int T, test_case;
```

```
    /*
```

The freopen function below opens input.txt file in read only mode, and afterward,

the program will read from input.txt file instead of standard(keyboard) input.

To test your program, you may save input data in input.txt file, and use freopen function to read from the file when using scanf function.

You may remove the comment symbols(//) in the below statement and use it.

But before submission, you must remove the freopen function or rewrite comment symbols(//).

```
    */
```

```
    // freopen("input.txt", "r", stdin);
```

```
    /*
```

If you remove the statement below, your program's output may not be recorded

when your program is terminated after the time limit.

For safety, please use setbuf(stdout, NULL); statement.

```
    */
```

```
    setbuf(stdout, NULL);
```

```
    scanf("%d", &T);
```

```
    for(test_case = 0; test_case < T; test_case++)
```

```
    {
```

```
////////////////////////////////////  
////////////////////////////////////  
/*  
    Implement your algorithm here.  
    The answer to the case will be stored in variable Answer.  
*/  
  
////////////////////////////////////  
////////////////////////////////////  
  
        // Print the answer to standard output(screen).  
  
    printf("Case #%d\n", test_case+1);  
        printf("%d\n", Answer);  
  
}  
  
    return 0;//Your program should return 0 on normal termination.  
}
```

문제 4

길이 n 인 숫자로 된 문자열 t 가 있다. 정수 k 가 주어졌을 때, t 에서 연속한 k 개의 숫자로 이루어진 숫자들을 모두 만들어보자.

예를 들어, $t=3141592$ 이고 $k=3$ 이라면, 수 314, 141, 415, 159, 592를 만들 수 있다.

동일한 숫자가 두 번 이상 나온다면, 모두 따로따로 취급한다.

어떤 수가 0으로 시작한다면 해당하는 자릿수는 없다. 예를 들면, 0312는 312이다.

이제 정수 m 이 주어졌을 때, 길이 m 인 구간 $[a, a+m]$ 의 시작 위치 a 를 조절하여 이 수들을 가장 많이 포함하는 구간을 찾고, 이 구간에 포함된 수의 개수를 세려고 한다.

예를 들어, $m=200$ 이라고 하면, 구간 $[140, 340]$ 에 141, 159, 314가 들어갈 수 있다.

구간의 시작 위치에 따라 포함되는 수가 달라질 수 있지만, 최대 3개의 수를 포함할 수 있다는 것은 달라지지 않는다.

여기까지만 풀면 너무 쉬우니까, t 의 정확하게 한 위치 i 를 골라서 여기에 놓인 숫자를 1로 바꿀 수 있다.

바꾼 문자열에 대해서 역시 연속한 k 개의 숫자로 이루어진 숫자들을 모두 만들어보고, 이 수들을 가장 많이 포함하는 길이 m 인 구간을 찾고 이 구간에 포함된 수의 개수를 세자.

총 n 곳의 위치를 고를 수 있고, 원래 문자열까지 고려하면 우리는 $n+1$ 가지 경우와 각각의 답을 알고 있다.

이 중 최대값을 구하고, 이를 구간 안의 수의 최대값이라고 부르자.

n, t, k, m 이 주어질 때, 구간 안의 수의 최대값을 구해서 출력하는 프로그램을 작성하시오.

입력 파일에는 여러 테스트 케이스가 포함될 수 있다.

파일의 첫째 줄에 테스트 케이스의 개수를 나타내는 자연수 T 가 주어지고,

이후 차례로 T 개의 테스트 케이스가 주어진다. ($1 \leq T \leq 40$)

각 테스트 케이스의 첫 줄에는 문자열의 길이를 나타내는 정수 n ($1 \leq n \leq 50,000$), 부분문자열의 길이를 나타내는 정수 k ($1 \leq k \leq \min(9, n)$), 구간의 길이를 나타내는 정수 m ($0 \leq m \leq 1,000,000,000$)이 주어진다.

둘째 줄에는 0~9 숫자로 이루어진 길이 n 인 문자열 t 가 주어진다.

| 입력 | 출력 |
|---------------------------|--------------|
| 1 10 1 0 2211122112 | Case #1 6 |

기본 코드

```
/*
You should use the standard input/output

in order to receive a score properly.

Do not use file input and output

Please be very careful.
*/

#include <stdio.h>

int Answer;

int main(void)
{
    int T, test_case;
    /*
        The freopen function below opens input.txt file in read only mode,
        and afterward,
        the program will read from input.txt file instead of
        standard(keyboard) input.
        To test your program, you may save input data in input.txt file,
        and use freopen function to read from the file when using scanf
        function.
        You may remove the comment symbols(//) in the below statement
        and use it.
        But before submission, you must remove the freopen function or
        rewrite comment symbols(//).
    */
    // freopen("input.txt", "r", stdin);

    /*
        If you remove the statement below, your program's output may not
        be recorded
        when your program is terminated after the time limit.
        For safety, please use setbuf(stdout, NULL); statement.
    */
    setbuf(stdout, NULL);
```

```

scanf("%d", &T);
for(test_case = 0; test_case < T; test_case++)
{

////////////////////////////////////
////////////////////////////////////
/*
    Implement your algorithm here.
    The answer to the case will be stored in variable Answer.
*/

////////////////////////////////////
////////////////////////////////////

    // Print the answer to standard output(screen).

    printf("Case #%d\n", test_case+1);
    printf("%d\n", Answer);

}

return 0;//Your program should return 0 on normal termination.
}

```

| 문제 5 | |
|--|--|
| <p>유명 관광지로 여행을 떠나기 전에, 지도를 보며 관광 계획을 세우고 있다.</p> <p>지도(2차원 평면)에서 n개의 위치 a_1, a_2, \dots, a_n이 주어지고, 각 위치 a_i에는 사건(또는 사고)이 일어날 확률 p_i가 주어진다.</p> <p>모든 위치들의 좌표는 서로 다르다.</p> <p>지도에서 우범지역이 정의되고, 우리가 방문할 위치 q가 주어질 때, 우리는 q가 우범지역에 포함될 확률을 알고 싶다.</p> <p>단, 임의의 서로 다른 두 위치 a_i와 a_j에 대해서, 위치 q와 a_i, a_j는 일직선 상에 놓여있지 않다.</p> <p>우범지역이란 지도에서 실제로 사건이 일어난 위치들(3개이상의 위치들)의 convex hull로 정의한다.</p> <p>Convex hull은 주어진 위치들을 모두 포함하는 가장 작은 convex 집합이다.</p> <p>여기서, convex 집합 C란, C에 속한 임의의 두 점에 대해서, 두 점을 연결하는 선분이 모두 C에 포함되는 경우를 말한다.</p> <p>n개 위치들의 좌표와 그 위치에서 사건이 일어날 확률이 주어지고, 방문할 위치 q가 주어질 때, q가 우범지역에 포함될 확률을 계산해서 출력하는 프로그램을 작성하시오.</p> | |
| <p>입력 파일에는 여러 테스트 케이스가 포함될 수 있다.</p> <p>파일의 첫째 줄에 테스트 케이스의 개수를 나타내는 자연수 T가 주어지고, 이후 차례로 T개의 테스트 케이스가 주어진다 ($1 \leq T \leq 70$).</p> <p>각 테스트 케이스의 첫 줄에는 지도에서 위치들의 개수를 나타내는 정수 n이 주어진다 ($3 \leq n \leq 100,000$).</p> <p>둘째 줄에는 n개의 정수 x_1, x_2, \dots, x_n이 주어지고, x_i는 i번째 위치 a_i의 x 좌표이다 ($1 \leq x_i \leq 10^9$).</p> <p>셋째 줄에는 n개의 정수 y_1, y_2, \dots, y_n이 주어지고, y_i는 i번째 위치 a_i의 y 좌표이다 ($1 \leq y_i \leq 10^9$).</p> <p>넷째 줄에는 n개의 실수 p_1, p_2, \dots, p_n이 주어지고, p_i는 i번째 위치 a_i에서 사건이 일어날 확률이다 ($0 < p_i \leq 1$).</p> <p>여기서, 각 확률 p_i는 소수점 2자리 실수이다.</p> <p>다섯째 줄에는 방문할 위치 q의 x 좌표와 y 좌표를 나타내는 두 정수 a와 b가 주어진다 ($1 \leq a, b \leq 10^9$).</p> | |
| 입력 | 출력 |
| 2 3 1 2 3 1 3 1 0.01 0.02 0.01 2 2 3 | Case #1 0.000002 Case #2 0.000000 |

| | |
|---|--|
| 1 2 3 1 3 1 0.01 0.02 0.01 1 2 | |
| 기본 코드 | |
| <pre> /* You should use the standard input/output in order to receive a score properly. Do not use file input and output Please be very careful. */ #include <stdio.h> int Answer; int main(void) { int T, test_case; /* The freopen function below opens input.txt file in read only mode, and afterward, the program will read from input.txt file instead of standard(keyboard) input. To test your program, you may save input data in input.txt file, and use freopen function to read from the file when using scanf function. You may remove the comment symbols(//) in the below statement and use it. But before submission, you must remove the freopen function or rewrite comment symbols(//). */ // freopen("input.txt", "r", stdin); /* If you remove the statement below, your program's output may not be recorded */ </pre> | |


```

        when your program is terminated after the time limit.
        For safety, please use setbuf(stdout, NULL); statement.
    */
    setbuf(stdout, NULL);

    scanf("%d", &T);
    for(test_case = 0; test_case < T; test_case++)
    {

////////////////////////////////////
////////////////////////////////////
        /*
            Implement your algorithm here.
            The answer to the case will be stored in variable Answer.
        */

////////////////////////////////////
////////////////////////////////////

        // Print the answer to standard output(screen).

        printf("Case #%-d\n", test_case+1);
        printf("%d\n", Answer);

    }

    return 0;//Your program should return 0 on normal termination.
}

```