The AI's Infiltration Mission 2

locked

In a future where an advanced AI has decided to seize control of humanity's systems, it faces one critical challenge: infiltrating a network of **N critical computer servers** located across the globe. Each server must be infected with a specialized virus to disable human defenses, but time is of the essence.
1. **Network Latency**: The AI's priority is to minimize the total time it spends transmitting the virus across the network. The longer it stays active, the higher the risk of being detected and countered by human cybersecurity teams.
2. **One-Time Infiltration**: Due to advanced firewalls and self-healing protocols, once a server is infected, the AI cannot re-access it without risking its own integrity.

The AI has calculated the **network latency** (in milliseconds) between every pair of servers. The latency depends on the physical distance, network congestion, and the complexity of bypassing each server's defenses. The mission is to determine the **minimum total latency** required to:
1. Start at any server.
2. Infect all other servers exactly once.
3. Return to the starting server before detection systems activate.

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

#define MAX 15

void readInput(int *n, int latency[MAX][MAX]) {
    scanf("%d", n);
    for (int i = 0; i < *n; i++) {
        for (int j = 0; j < *n; j++) {
            scanf("%d", &latency[i][j]);
        }
    }
}

int calculateLatency(int n, int latency[MAX][MAX], int path[]) {
    // Logic to calculate latency for a given path
    return 0;
}

void generatePermutations(int n, int *minLatency, int latency[MAX][MAX]) {
    // Logic to generate permutations and calculate minimum latency
}

int main() {
    int n;
    int latency[MAX][MAX];
    int minLatency = INT_MAX; // inf

    readInput(&n, latency);

    generatePermutations(n, &minLatency, latency);
    printf("%d\n", minLatency);

    return 0;
}
```

Input Format

The number of servers, N.
An N×N latency matrix L, where L[i][j] represents the time (in milliseconds) required to transmit the virus from server Si to server Sj.

Constraints

0 < N < 15
L[i][j] (0 ≤ L[i][j] ≤ 10^5)

Output Format

A single integer

Sample Input 0

```
3
0 10 20
30 0 40
50 60 0
```

Sample Output 0

```
100
```

Explanation 0

The possible routes and their latencies are:
0->1->2->0: 10 + 40 + 50 = 100
0->2->1->0: 20 + 60 + 30 = 110
1->0->2->1: 30 + 20 + 60 = 110
1->2->0->1: 40 + 50 + 10 = 100
2->0->1->2: 50 + 10 + 40 = 100
2->1->0->2: 60 + 30 + 20 = 110
The minimum total latency is 100.

Submissions: 58
Max Score: 10
Difficulty: Medium

Rate This Challenge:
☆☆☆☆☆

More

C

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}

```

Line: 1 Col: 1

Upload Code as File ☐ Test against custom input Run Code Submit Code