

[Nhà của tôi](#) / [Các khoá học của tôi](#) / [LTDI HK1 2526 \(GV PXHIEN\)](#) / [Thực hành](#) / [Buổi 1](#)

Câu hỏi **11**

Đúng

Đạt điểm 1,00

Cho cấu trúc dữ liệu đồ thị được khai báo sử dụng ma trận đỉnh - đỉnh như sau:

```
typedef struct {
    int A[100][100];
    int n;
} Graph;
```

Giả sử đồ thị vô hướng, không chứa khuyên, nhưng có chứa đa cung

Viết hàm **int edgeCount(Graph* G)** để đếm số cung của đồ thị G.

```
int edgeCount(Graph* G) {
}
```

For example:

Input	Result
<pre>4 0 2 1 0 2 0 3 0 1 3 0 1 0 0 1 0</pre>	7

Answer: (penalty regime: 0 %)

```
1 // int edgeCount(Graph* G) {
2 //     int i,j,count=0;
3 //     for(i=1;i<=G->n;i++){
4 //         for(j=1;j<=G->n;j++){
5 //             count+=G->A[i][j]; //số bậc
6 //         }
7 //     }
8 //     return count/2;
9 // }
10
11 int edgeCount(Graph* G) {
12     int count = 0;
13     for (int u = 1; u <= G->n; u++) {
14         for (int v = u + 1; v <= G->n; v++) {
15             count += G->A[u][v]; // số cung giữa u và v
16         }
17     }
18     return count;
19 }
20
```

Chọn

Debug: source code from all test runs

Run 1

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

typedef struct {
    int A[100][100];
    int n;
} Graph;

// int edgeCount(Graph* G) {
//     int i,j,count=0;
//     for(i=1;i<=G->n;i++){
//         for(j=1;j<=G->n;j++){
//             count+=G->A[i][j]; //số bậc
//         }
//     }
//     return count/2;
// }

int edgeCount(Graph* G) {
    int count = 0;
    for (int u = 1; u <= G->n; u++) {
        for (int v = u + 1; v <= G->n; v++) {
            count += G->A[u][v]; // số cung giữa u và v
        }
    }
    return count;
}

int main() {
    Graph G;
    scanf("%d", &G.n);
    int i, j;
    for (i = 1; i <= G.n; i++)
        for (j = 1; j <= G.n; j++) {
            scanf("%d", &G.A[i][j]);
        }

    printf("%d\n", edgeCount(&G));

    ;

    return 0;
}
```

Run 2

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#include <stdbool.h>
#include <math.h>

typedef struct {
    int A[100][100];
    int n;
} Graph;

// int edgeCount(Graph* G) {
//     int i,j,count=0;
//     for(i=1;i<=G->n;i++){
//         for(j=1;j<=G->n;j++){
//             count+=G->A[i][j]; //số bậc
//         }
//     }
//     return count/2;
// }

int edgeCount(Graph* G) {
    int count = 0;
    for (int u = 1; u <= G->n; u++) {
        for (int v = u + 1; v <= G->n; v++) {
            count += G->A[u][v]; // số cung giữa u và v
        }
    }
    return count;
}

int main() {
    Graph G;
    scanf("%d", &G.n);
    int i, j;
    for (i = 1; i <= G.n; i++)
        for (j = 1; j <= G.n; j++) {
            scanf("%d", &G.A[i][j]);
        }

    printf("%d\n", edgeCount(&G));

    ;

    return 0;
}

```

	Input	Expected	Got	
✓	4 0 2 1 0 2 0 3 0 1 3 0 1 0 0 1 0	7	7	✓
✓	5 0 1 2 0 3 1 0 0 0 0 2 0 0 1 1 0 0 1 0 1 3 0 1 1 0	9	9	✓

Passed all tests! ✓

◀ tt

Chuyển tới...

Buổi 2 - phần cơ bản ▶