

[Dashboard](#) / [My courses](#) / [Graph Theory-HK3-0405](#) / [Tuần 8 - Thứ tự topo & Ứng dụng](#) / [Chia kẹo](#)

Started on	Tuesday, 1 July 2025, 10:45 PM
State	Finished
Completed on	Tuesday, 1 July 2025, 10:45 PM
Time taken	12 secs
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

Cô giáo Trang chuẩn bị kẹo để phát cho các bé mà cô đang giữ. Dĩ nhiên mỗi bé đều có một tên gọi rất dễ thương ví dụ: Mạnh Phát, Diễm Quỳnh, Đăng Khoa, ... Tuy nhiên, để đơn giản vấn đề ta có thể giả sử các em được đánh số từ 1 đến n .

Cô giáo muốn rằng tất cả các em đều phải có kẹo. Cô lại biết thêm rằng có một số bé có ý muốn hơn bạn mình một chút vì thế các em ấy muốn kẹo của mình nhiều hơn của bạn.

Hãy viết chương trình giúp cô tính xem mỗi em cần được chia ít nhất bao nhiêu kẹo và tổng số kẹo ít nhất mà cô phải chuẩn bị là bao nhiêu.

Đầu vào (Input):

Dữ liệu đầu vào được nhập từ bàn phím với định dạng:

- Dòng đầu tiên chứa 2 số nguyên n và m , tương ứng là số bé và số cặp bé mà trong đó có 1 bé muốn có kẹo hơn bạn mình.
- m dòng tiếp theo mỗi dòng chứa 2 số nguyên a, b nói rằng bé a muốn có kẹo nhiều hơn bé b .

Đầu ra (Output)

- In ra màn hình số kẹo ít nhất của từng em, mỗi em trên một dòng.
- Dòng cuối cùng in tổng số kẹo ít nhất mà cô giáo Trang cần phải chuẩn bị.

Xem thêm các ví dụ để hiểu thêm về đầu vào và đầu ra.

For example:

Input	Result
3 2	3
1 3	1
3 2	2
	6

Answer: (penalty regime: 10, 20, ... %)

```

1 #include <stdio.h>
2
3 #define MAX_N 100
4 typedef struct {
5     int n, m;
6     int A[MAX_N][MAX_N];
7 } Graph;
8
9
10 void init_graph(Graph *pG, int n) {
11     pG->n = n;
12     pG->m = 0;
13     for (int u = 1; u <= n; u++)
14         for (int v = 1; v <= n; v++)
15             pG->A[u][v] = 0;
16 }
17
18 void add_edge(Graph *pG, int u, int v) {
19     pG->A[u][v] += 1;
20 }
21
22

```

	Input	Expected	Got	
✓	3 2 1 3 3 2	3 1 2 6	3 1 2 6	✓
✓	7 10 2 1 3 1 4 1 3 2 6 2 7 3 5 4 3 5 7 5 7 6	1 2 4 2 3 3 5 20	1 2 4 2 3 3 5 20	✓

Passed all tests! ✓

Question author's solution (C):

```
1 #include <stdio.h>
2
3 #define MAX_N 100
4 typedef struct {
5     int n, m;
6     int A[MAX_N][MAX_N];
7 } Graph;
8
9
10 void init_graph(Graph *pG, int n) {
11     pG->n = n;
12     pG->m = 0;
13     for (int u = 1; u <= n; u++)
14         for (int v = 1; v <= n; v++)
15             pG->A[u][v] = 0;
16 }
17
18 void add_edge(Graph *pG, int u, int v) {
19     pG->A[u][v] += 1;
20 }
21
22
```

Correct

Marks for this submission: 1.00/1.00.

[◀ Tổ chức thi công - Dự án phần mềm](#)

Jump to...

[Xếp hạng đồ thị \(check được\) ▶](#)