# **Những người lái xe buýt Tên file: BUSDRIVER.\***

Trong một thành phố có ***n*** tài xế xe buýt. Ngoài ra còn có ***n*** tuyến xe buýt ban ngày và các tuyến xe buýt ban tối với chiều dài khác nhau. Mỗi tài xế được chỉ định một tuyến đường ban ngày và một tuyến đường ban tối. Đối với bất kỳ tài xế nào, nếu tổng chiều dài tuyến của anh ta vượt quá ***d*** giờ, anh ta phải được trả lương làm thêm giờ cho mỗi giờ sau ***d*** giờ đầu tiên với một giờ ***r*** đồng / giờ.

Nhiệm vụ của bạn là chỉ định một tuyến đường ban ngày và một tuyến đường ban tối cho mỗi tài xế xe buýt để tổng số tiền làm thêm giờ mà công ty phải trả cho họ càng nhỏ càng tốt.

**INPUT: BUSDRIVER.INP**

* Gồm nhiều bộ test, mỗi bộ test có dạng như sau:
  + Dòng đầu tiên chứa 3 số nguyên ***n***, ***d***, ***r*** – được mô tả như trên
  + Dòng thứ hai chứa ***n*** số nguyên (có giá trị không lớn hơn 10000) thể hiện độ dài của ***n*** tuyến xe buýt ban ngày.
  + Dòng thứ ba chứa ***n*** số nguyên (có giá trị không lớn hơn 10000) thể hiện độ dài của ***n*** tuyến xe buýt ban tối.
* Kết thúc input là 3 số 0.

**OUTPUT: BUSDRIVER.OUT**

* Gồm nhiều dòng, mỗi dòng là số tiền phải trả nhỏ nhất cho các lái xe buyt tương ứng với các test trong input.
* Giới hạn:
  + 1 ≤ ***n*** ≤ 100
  + 1 ≤ ***d*** ≤ 10000
  + 1 ≤ ***r*** ≤ 5

***Ví dụ:***

|  |  |
| --- | --- |
| **BUSDRIVER.INP** | **BUSDRIVER.OUT** |
| 2 20 5  10 15  10 15  2 20 5  10 10  10 10  3 4 2  4 2 3  1 3 2  0 0 0 | 50  0  6 |

# **Bảng chữ cái Tên file: ALPHABET.\***

Trong bài toán này, bạn phải vẽ một hình vuông bằng cách sử dụng bảng chữ cái tiếng Anh viết hoa.

Để chính xác hơn, bạn sẽ được cung cấp một hình vuông với một số khối trống và các khối khác đã được điền sẵn cho bạn với một số chữ cái để làm cho công việc của bạn dễ dàng hơn. Bạn phải chèn ký tự vào mọi ô trống để toàn bộ lưới được lấp đầy bằng bảng chữ cái. Khi làm như vậy bạn phải đáp ứng các quy tắc sau:

1. Đảm bảo không có ô liền kề nào chứa cùng một chữ cái; hai ô liền kề nếu chúng có chung cạnh.

2. Có thể có nhiều cách để lấp đầy lưới điện. Bạn phải đảm bảo bạn điền vào bảng sao cho các chữ cái có thứ tự từ điển nhỏ nhất. Ở đây, hai lưới được kiểm tra theo thứ tự hàng khi so sánh theo từ điển.

INPUT: ALPHABET.INP

* Dòng đầu tiên chứa số nguyên dương T, số lượng bộ test
* Mỗi bộ test có dạng:
  + Dòng đầu chứa số nguyên dương n (n ≤ 20), kích thước của bảng hình vuông.
  + n dòng tiếp theo, mỗi dòng chứa n ký tự ‘.’ (thể hiện ô trống) hoặc ký tự nào đó thuộc đoạn [‘A’, ‘Z’] (thể hiện ô đã được điền sẵn).

OUTPUT: ALPHABET.OUT

* Tương ứng mỗi test, tại test thứ i ghi ra từ “Case i:’
* Các dòng tiếp theo ghi ra bảng chữ cái thoản mãn yêu cầu đề bài

Ví dụ:

|  |  |
| --- | --- |
| ALPHABET.INP | ALPHABET.OUT |
| 2  3  ...  ...  ...  3  ...  A..  ... | Case 1:  ABA  BAB  ABA  Case 2:  BAB  ABA  BAB |

# **GỌI MÓN ĂN Tên file: CALLFOOD.CPP**

Bình An đang rất đói và cô quyết định ăn trưa tại căng tin nhà trường. Căng tin có món ăn khác nhau và thật đặc biệt: món ăn thứ có hai giá và trong đó khách hàng phải trả nếu là món ăn đầu tiên được gọi trong bữa, các trường hợp còn lại món có giá .

Vì rất đói nên Bình An không thể quyết định là chọn những món nào để ăn. Cô ta quyết định hỏi bạn rằng nếu ăn đúng món thì phải trả số tiền ít nhất là bao nhiêu?

### Dữ liệu vào:

* Dòng đầu tiên ghi số nguyên dương là số lượng các món ăn.
* dòng tiếp theo, dòng thứ chứa hai số nguyên dương và là giá của món thứ theo mô tả ở trên.

### Kết quả ra:

* Ghi dòng, dòng thứ ghi số tiền tối thiểu phải trả khi ăn đúng món ăn trong số món ăn của căng tin.

Ví du:

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| 3  14 5  9 3  10 5 | 9  13  18 |
| 5  1000000000 1000000000  1000000000 1000000000  1000000000 1000000000  1000000000 1000000000  1000000000 1000000000 | 1000000000  2000000000  3000000000  4000000000  5000000000 |

**\* Giải thích ví dụ 1:**

- k = 1: chọn món ăn thứ 2, tổng tiền bằng 9

- k = 2: chọn món ăn thứ 2 và 3, tổng tiền bằng 10 + 3 = 13

- k = 3: chọn cả ba món ăn, tổng tiền bằng 10 + 3 + 5 = 18

**\* Ràng buộc:**

* Subtask1: 30% số test tương ứng với
* Subtask2: 30% số test tiếp theo ứng với tất cả các phần tử của dãy A và dãy B bằng nhau
* Subtask3: 40% số test còn lại không có ràng buộc gì