# ĐỀ NGÀY 9 - 12 - 2022

# Bài 1. DINNER

Nhóm *k* sinh viên trong trường trung cấp Nấu ăn quyết định sẽ cùng nhau tự nấu ăn tối trong suốt học kỳ gồm *n* ngày. Đó vừa là cách tiết kiệm tiền, vừa là cách thực hành!

Để đảm bảo công bằng, mọi người thống nhất là mỗi người phải ít nhất một lần nấu buổi tối cho cả nhóm. Cần lên lịch để mọi người chủ động trong công việc. Mọi người quan tâm là có thể có bao cách lên lịch khác nhau. Các sinh viên được đánh số từ 1 đến *k*, các ngày trong học kỳ - đánh số từ 1 đến *n*. Với *k* = 2 và *n* = 3 ta có thể có 6 cách: (1*;* 1*;* 2), (1*;* 2*;* 1), (2*;* 1*;* 1), (1*;* 2*;* 2), (2*;* 1*;*2), (2*;* 2*;* 1).

**Yêu cầu:** Cho *k* và *n*. Hãy xác định số cách lên lịch khác nhau.

# Input

* Gồm một dòng chứa 2 số nguyên *k* và *n* (1 ≤ *k* ≤ *n* ≤ 100).

# Output

* Số lượng lịch tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **DINNER.INP** | **DINNER.OUT** |
| 2 3 | 6 |

# Bài 2. Mảnh giấy

Alice là một giáo viên lớp mầm, hôm nay Alice muốn dạy các bé học tô màu. Alice chuẩn bị một mảnh giấy kích cỡ 𝑁 × 𝑀 được chia thành 𝑁 × 𝑀 ô vuông nhỏ 1 × 1. Có 𝐶 trẻ trong lớp của Alice được đánh số từ 1 đến 𝐶, mỗi trẻ sẽ có một bút tô màu với các màu khác nhau và sẽ lần lượt tô một hình chữ nhật con nằm trong mảnh giấy của Alice đưa cho ban đầu. Alice muốn cắt ra tất cả các ô vuông nhỏ 1 × 1 mà được tô màu bởi cả 𝐶 trẻ để treo lên tường trang trí, nhưng Alice muốn số lượng các ô vuông như vậy phải ít nhất là 𝐾 (𝐾 ≤ 𝑁 ∗ 𝑀).

Alice muốn biết có bao nhiêu cách tô màu mà thỏa mãn điều kiện trên. Hai cách tô màu khác nhau khi tồn tại một trẻ mà trong hai cách tô màu đó tô màu hai hình chữ nhật con khác nhau.

# Input

* Dòng đầu tiên: số nguyên 𝑇 là số bộ test (𝑇 ≤ 5);
* 𝑇 dòng sau mỗi dòng gồm 4 số nguyên dương 𝐶, 𝑁, 𝑀, 𝐾 (𝑁, 𝑀 ≤ 1000, 𝐶 ≤ 105) lần lượt là số trẻ, kích cỡ mảnh giấy và số ô vuông ít nhất mà Alice mong muốn.

# Output

* Gồm 𝑇 dòng, mỗi dòng là kết quả bài toán lấy dư cho 109 + 7.

Ví dụ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **paper.inp** | | | | **paper.out** |
| 1  2 | 2 | 3 | 4 | 7 |

# Bài 3. Trò chơi trí tuệ

Chọn một nhóm gồm 𝑛 người ngồi vào 𝑛 vị trí, các vị trí được đánh số từ 1 đến 𝑛 theo chiều kim đồng hồ quanh bàn tròn. Người ngồi ở vị trí 𝑖 gọi là người thứ 𝑖 (𝑖 = 1, 2, … , 𝑛), như vậy, với người thứ nhất, bên phải là người thứ 𝑛, bên trái là người thứ 2; với người thứ 2 bên phải là người thứ nhất, bên trái là người thứ 3; ...; còn với người thứ 𝑛, bên phải là người thứ (𝑛 − 1), bên trái là người thứ nhất. Người thứ 𝑖 được phát 𝑎𝑖 (𝑎𝑖 ≥ 0) viên kẹo. Mỗi lượt, chỉ một người (có số lượng kẹo lớn hơn 0) được phép chuyển một viên của mình cho người bên trái hoặc người bên phải. Nhóm sẽ giành chiến thắng và nhận được phần thưởng của Ban tổ chức nếu sau khi thực hiện dãy các lượt chuyển kẹo thì có không quá một người có số kẹo là một số lẻ. Một số thầy, cô dạy môn Tin học nhanh chóng nhận thấy, nếu biết thời gian để chuyển một viên kẹo sang cho người bên trái là 𝐿 và sang cho người bên phải là 𝑅 thì có thể tính toán chính xác thời gian ít nhất để nhóm giành chiến thắng.

**Yêu cầu:** Cho số nguyên dương 𝑛 số nguyên không âm 𝑎1, 𝑎2, … , 𝑎𝑛 và hai số nguyên dương 𝐿, 𝑅. Bạn hãy lập trình tính thời gian ít nhất để nhóm giành chiến thắng.

**Dữ liệu:** Vào từ file văn bản SWGAME.INP theo khuôn dạng:

* Dòng đầu tiên chứa ba số nguyên dương 𝑛, 𝐿, 𝑅 (𝐿, 𝑅 ≤ 106);
* Dòng thứ hai chứa 𝑛 số nguyên không âm 𝑎1, 𝑎2, … , 𝑎𝑛 (𝑎𝑖 ≤ 106) là số kẹo mà người thứ 𝑖 được phát.

**Kết quả:** Ghi ra file văn bản SWGAME.OUT một số nguyên là thời gian ít nhất để nhóm giành chiến thắng.

***Ví dụ:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SWGAME.INP** | | | | | **SWGAME.OUT** |
| **5** | **3** | **2** |  |  | **2** |
| **1** | **2** | **3** | **4** | **5** |  |

# Ràng buộc:

* Có 25% số test ứng với 25% số điểm của bài có 𝑛 ≤ 10;
* Có 50% số test ứng khác với 25% số điểm của bài có 𝑛 ≤ 1000;
* Có 25% số test còn lại ứng với 25% số điểm của bài có 𝑛 ≤ 105.

# Bài 4. Thử nghiệm robot

Công ty AZ đang sản xuất robot vận chuyển hàng hóa tự động trên mặt phẳng. Để làm việc đó, công ty tiến hành huấn luyện các robot trên một đường băng phẳng được chia thành 𝑛 chặng, đánh số từ 1 đến 𝑛 theo chiều từ trái sang phải. Chặng thứ 𝑖 sẽ có một chiếc hộp màu 𝑐𝑖 (𝑖 = 1,2, … , 𝑛) và loại hộp màu 𝑐𝑖 sẽ có khối lượng 𝑤𝑐𝑖 kg. Robot có khả năng vận chuyển 𝑘 kg, robot xuất phát từ chặng thứ nhất, di chuyển và kết thúc tại chặng thứ 𝑛. Khi robot ở một chặng, robot có thể thực hiện một

trong các hành động sau:

1. Tiếp tục di chuyển sang chặng tiếp theo nếu chặng hiện tại chưa phải là chặng kết thúc (chặng thứ 𝑛);
2. Lấy chiếc hộp ở chặng hiện tại cho vào giỏ hàng nếu khi cho vào tổng số khối lượng các hộp trong giỏ không vượt quá 𝑘, tiếp theo thực hiện hành động 1;
3. Bỏ một chiếc hộp ở trong giỏ ra nếu chiếc hộp đó cùng màu với hộp ở chặng hiện tại, tiếp theo thực hiện hành động 1;
4. Bỏ một chiếc hộp ở trong giỏ ra nếu chiếc hộp đó cùng màu với hộp ở chặng hiện tại, tiếp theo thực hiện hành động 2.

**Yêu cầu:** Cho biết thông tin các hộp trên đường băng, khối lượng của từng loại hộp màu và số nguyên

𝑘, hãy xác định số lượng nhiều nhất các hộp có thể di chuyển.

# Input

* Gồm nhiều bộ dữ liệu, mỗi bộ gồm hai dòng theo khuôn dạng:
  + Dòng thứ đầu tiên chứa số nguyên 𝑛, 𝑘, 𝑑 (𝑘 ≤ 106; 𝑑 ≤ 26);
  + Dòng thứ hai chứa một xâu độ dài 𝑛 chỉ gồm 𝑑 kí tự đầu tiên trong bảng chữ cái thường mô tả màu của các hộp;
  + Dòng thứ ba gồm 𝑑 số nguyên mô tả cân nặng của từng loại hộp màu, liệt kê theo thứ tự trong bảng chữ cái.

# Output

* Gồm nhiều dòng, mỗi dòng là kết quả tương ứng với bộ dữ liệu vào là số lượng nhiều nhất các hộp có thể di chuyển.

|  |  |
| --- | --- |
| **CWBROBOT.INP** | **CWBROBOT.OUT** |
| 6 2 3  abcbca 1 2 1 | 2 |
| 5 1 1  aaaaa 1 | 4 |

**Subtask 1:** 𝑛 ≤ 20;

**Subtask 2:** 𝑛 ≤ 200; 𝑘 = 3;

**Subtask 3:** 𝑛 ≤ 200; 𝑑 ≤ 10;

**Subtask 4:** 𝑛 ≤? ? ? ; các hộp màu có cùng khối lượng.