|  |  |
| --- | --- |
| Bắc Giang PreVOI  Ngày thi thứ nhất | ĐỀ THI  *Thời gian: 180 phút* |

Tổng quan đề thi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File mã nguồn | File input | File output | Thời gian | Bộ nhớ |
| Oddcoin.[cpp|pas] | Oddcoin.inp | Oddcoin.out | 1s | 512MB |
| Mine.[cpp|pas] | Mine.inp | Mine.out | 0.4s | 512MB |
| Construct.[cpp|pas] | Construct.inp | Construct.out | 2s | 512MB |

Hãy lập trình giải 3 bài sau đây.

## 1. ODDCOIN

Nước ODD có các đồng xu mệnh giá sau 1, 3, 5, 10, 30, 50, 100, 300, 500, 1000, 3000, 5000, 10 000, 30 000 và 50 000 đồng. Ngân hàng ban hành thêm 1 loại xu mới với mệnh giá x đồng. Để trả đúng t đồng, cần tối thiểu bao nhiêu đồng xu?

INPUT

Dòng đầu ghi số test q. q dòng sau mỗi dòng ghi 2 số x, t.

OUTPUT

In ra q dòng, mỗi dòng là kết quả một truy vấn

GIỚI HẠN

* q ≤ 50 và xi khác ti
* 25% số test có : x ≤ 20, ti ≤ 20
* 25% số test có : x ≤ 105, ti ≤ 105
* 25% số test có : x ≤ 5.107, ti ≤ 5.107
* 25% số test có : x ≤ 2.109, ti ≤ 2.109

|  |  |
| --- | --- |
| oddcoin.inp | oddcoin.out |
| 4  4700 53  4700 9400  4700 9401  4700 30000 | 2  2  3  1 |

## 2. MINE

Một nhóm nhân viên khai mỏ có dự định tiến vào một hầm mỏ dài và rất hẹp. Vì rất hẹp và dài, nên chỉ người vào cuối cùng mới có thể rời khỏi hầm mỏ. Ban đầu, hầm mỏ trống. Ở cửa hầm mỏ, chúng ta ghi lại thời gian một ai đó vào và ra hầm mỏ. Mỗi người thợ mỏ không được ở trong hầm mỏ quá m đơn vị thời gian. Hãy đếm xem có bao nhiêu tình huống khác nhau có thể xảy ra. Hai tình huống được xem là khác nhau, nếu tồn tại 1 thời điểm nào đó mà trong tình huống này, người i có trong hầm mỏ, trong tình huống khác, người i không có trong hầm mỏ. Các công nhân khai mỏ được đánh số từ 1 đến n. Công nhân khai mỏ thứ i không được vào hầm mỏ trước công nhân khai mỏ thứ i – 1 (Công nhân 1 là người vào đầu tiên).

INPUT

Dòng đầu ghi 2 số nguyên n và m, trong đó 1 ≤ n ≤ 200 000 và 1 ≤ m ≤ 1 000 000; Dòng thứ 2 ghi 2n số nguyên dương 1 ≤ a1 < a2 < … < a2n ≤ 1 000 000 là thời điểm (tính từ đầu ngày) có 1 người thợ mỏ vào hoặc ra hầm mỏ

OUTPUT

In ra kết quả theo mod 1 000 003.

GIỚI HẠN

* 10% số test có n ≤ 10
* 10% số test có n ≤ 200, m = 1,000,000
* 30% số test có n ≤ 200
* 40% số test có n ≤ 2,000
* 10% số test có n ≤ 200,000, m = 1,000,000

|  |  |
| --- | --- |
| mine.inp | mine.out |
| 3 6 1 2 3 7 9 10 | 2 |

Có 3 người thợ mỏ.

Giả sử người đầu (A) vào mỏ ở thời điểm 1.

Nếu A rời mỏ ở thời điểm 2, thì người 2 (B) sẽ vào mỏ ở thời điểm 3. Do không được ở trong mỏ ở quá 6, nên B không thể rời mỏ ở thời điểm 10. B không thể rời mỏ ở thời điểm 9 vì khi đó ở thời điểm 7 có người thứ 3 đi vào và chặn lỗi ra. Do đó B chỉ có thể rời ở thời điểm 7. Và người thứ 3 đến và đi ở thời điểm 9, 10. Đây là tình huống 1.

Nếu A rời mỏ ở thời điểm 7, thì ( 2, 3) và (9, 10) là 2 thời điểm đến và đi của 2 người khác nhau, đây là tình huống 2.

A không thể rời mỏ ở thời điểm 3 (vì thời điểm 2 có người đến và phong tỏa không cho A ra).

Vậy chỉ có thể có 2 tình huống.

## 3. CONSTRUCT

Trên đường phố sẽ được xây n tòa nhà nằm cạnh nhau, nhìn từ trái qua phải, độ cao dự kiến của chúng lần lượt là a1, a2,…., an.

Ban đầu, độ cao hi của các tòa nhà bằng 0.

Giám đốc H là tổng phụ trách. Ở mỗi giai đoạn, giám đốc H chọn 1 đoạn [L, R] để làm việc. Trong giai đoạn này, khi độ cao các tòa nhà hL, hL+1,…, hR sẽ tăng lên 1. Khi hi = ai với mọi 1 ≤ i ≤ n, dự án sẽ dừng.

Kế hoạch xây dựng luôn thay đổi, có m sự kiện thuộc 2 kiểu sau:

* 1 L R k (1 ≤ L ≤ R ≤ n,1 ≤ k ≤ 105) thay đổi ax thành ax + k với mọi L ≤ x ≤ R
* 2 L R (1 ≤ L ≤ R ≤ n), giả sử a1,a2,…,aL−1,aR+1,aR+2,…,an=0, hỏi xem cần tối thiểu bao nhiêu giai đoạn để dự án hoàn thành.

INPUT

Dòng đầu ghi số test T (1≤T≤1000). Ở mỗi test: dòng đầu ghi 2 số n và m (1≤n,m≤100000) là số tòa nhà và số sự kiện. Dòng thứ 2 ghi n số nguyên a1,a2,...,an (1≤ai≤100000). Sau đó là m dòng, mỗi dòng ghi m sự kiện như mô tả ở trên. Đảm bảo ∑n≤106 và ∑m≤106

OUTPUT

Với mỗi truy vấn, in ra trên 1 dòng kết quả cần tìm

SUBTASK

* Sub 1: (30%) ∑n <=1000, ∑m <=1000
* Sub2: (30%) ∑n <=1000, ∑m <=50000
* Sub3: (40%): Không có giới hạn gì thêm.

|  |  |
| --- | --- |
| construct.inp | construct.out |
| 1  5 4  1 3 1 4 5  2 1 5  1 3 4 2  2 2 4  2 1 5 | 7  6  6 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Hết\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_