**Група Y**

**ДЪРВЕСНИ ГЪБИ**

Учениците от СМГ се събрали в гората на Тенев баир на лов за дървесни гъби. Ще считаме, че гората представлява правоъгълна таблица с N реда и M стълба и във всяка нейна клетка има дърво с дадена височина (редовете са номерирани с числата от 0 до N-1 – от горе надолу, а стълбовете – с числата от 0 до M-1 от ляво надясно). Те търсят рядка дървесна гъба, която расте само по върховете на дърветата. За да я намерят трябва да се качат до върха на дадено дърво и да проверят дали тя е там. Освен това, докато са горе те могат да видят дали тя е на върха на някое от четирите съседни дървета, стига то да е по-ниско (строго) от дървото на което се намират те.

Тъй като учениците не искат да си губят времето, те ще проверяват само дървета, които са заобиколени от 4 по-ниски (така ще проверяват по 5 дървета наведнъж). Освен това, те търсят най-ниското такова дърво, за да не губят много време в катерене. Ако все още има няколко такива варианта те предпочитат дървото, което е най-близо до входа на парка, който е в горния ляв ъгъл - на координати (0, 0). Под най-близо разбираме с най-малко манхатаново разстояние: клетка с координати i и j, където i е редът, а j – колоната е на разстояние (i+j) от (0, 0). Ако има няколко най-близки отговарящи на описаните условия избираме това на най-горен ред (с минимално i).

От време на време в гората се случват промени: някое дърво пораства или отсичат върха на дърво. Учениците разбират всеки път когато настъпи такава промяна. Освен това ще считаме че шанса гъбата да е на всяко дърво е еднакъв (дори и на вече проверени дървета), тъй като тя много често се появява и изчезва.

Вашата задача е да помогнете на учениците да разберат на кое дърво е най-добре да се качат след всяка промяна в гората.

**Input Format**

* На първия ред се въвежда числото T – броя тестове. За всеки тест:
* На първия ред се въвеждат числата N, М и Q – броя редове на таблицата, броя колони на таблицата и броя промени в гората (заявки).
* Следващите N реда съдържат по M числа разделени с интервали - височините на дърветата на съответния ред от таблицата.
* Всеки от следващите Q реда съдържа по 3 числа, ред, колона и нова височина на дървото, което се променя.

**Constraints**

* 1 ≤ T ≤ 5
* 3 ≤ N, M ≤ 500
* 1 ≤ Q ≤ 50000
* Височините на дърветата са цели положителни числа < 109

**Output Format**

За всеки тест се извеждат по Q реда, като на всеки от тях пише реда и колоната на търсеното в задачата дърво разделени с интервал, след съответната заявка. Ако такова дърво (чиито 4 съседа са строго по-малки от него) не съществува: за съответната заявка да се изведе -1.

**Sample Input 0**

1

6 7 10

6 8 5 6 8 1 4

4 2 7 2 5 8 6

5 3 9 4 4 4 6

1 8 4 1 6 9 9

4 9 8 7 7 9 4

6 2 7 1 8 7 4

0 0 8

1 4 8

2 1 5

1 2 8

2 1 9

5 1 8

0 3 2

3 1 9

0 6 8

2 4 9

**Sample Output 0**

1 5

2 2

2 2

2 2

4 1

4 1

4 1

-1

-1

2 4

**Explanation 0**

Ще изредим координатите на дърветата с 4 по-ниски съседа след всяка заявка:

1. (1,5) (2,2) (4,1) - най-ниското е **(1,5)** с височина 8
2. (2,2) (4,1) - и двете са с височина 9, но **(2,2)** е по-близо до (0,0)
3. (2,2) (4,1) - и двете са с височина 9, но **(2,2)** е по-близо до (0,0)
4. (2,2) (4,1) - и двете са с височина 9, но **(2,2)** е по-близо до (0,0)
5. **(4,1)** - единствено
6. **(4,1)** - единствено
7. **(4,1)** - единствено
8. няма дървета удовлетворяващи условието т.е. **-1**
9. няма дървета удовлетворяващи условието т.е. **-1**
10. **(2,4)** - единствено

**МЕБЕЛИ**

ПрАси и приятелката и ПрЕси (накратко А и Е) се оказали в склад за мебели с тежката задача да изнесат всички мебели от там (добрата новина е че си остават за тях). Мебелите са наредени в N реда като в i-тия ред има Ki мебели. А и Е могат да изнасят по един мебел наведнъж. А може да изнася мебелите разположени в левите краища на редовете, докато Е може да изнася само тези в десните краища на редовете. Мебелите си имат цена и двете искали да вземат мебели на възможно най-висока цена (има и мебели с отрицателна цена). За това се споразумяли да играят на следната игра. Първо А взима един мебел отляво. После Е взима един мебел отдясно, после пак А отляво и после пак Е отдясно. Продължават по този начин докато не свършат мебелите. Двете се оказали доста умни и можели да играят оптимално. Пита се мебели на каква стойност ще задигне Праси при оптимална игра.

Забележка: Дори на всички леви краища да стоят мебели с отрицателна стойност, Праси е длъжна да играе ход и да вземе мебел с отрицателна цена, по който начин си намалява сумата.

**Input Format**

На първия ред се въвежда T - брой тестове. За всеки тест се въвежда N последвано от описание на N-те реда. Всеки ред се описва от Ki и на отделен ред, самите стойности на мебелите (в ред отляво надясно).

**Constraints**

1 <= T <= 10

стойностите на мебелите са в интервала [-10^9, 10^9]

Тест 1 (20 точки): N=2 и 1 <= Ki <= 50

Тест 2 (80 точки): K1 + К2 + ... + Кn <= 100 000

**Output Format**

На отделни редове изведете отговорите за всеки тест, а именно каква сума може да изкара Праси при оптимална игра от нейна страна както и на противника и.

**Sample Input 0**

1

2

5

1 2 1 2 1

5

2 1 2 1 2

**Sample Output 0**

8

**Explanation 0**

Борбата в този случай е за долния ред защото той е по-хубав (има по-ценни мебели). Най-добрите ходове на Праси са да взима докато може от долния ред и после каквото остане от горния. Преси също ще се бори за долния ред, обаче понеже Стаси е първа тя ще вземе 3 от долния ред а Преси само 2. После обаче, Преси е на ход и ще си разделят горния ред 2 към 3 в полза на Преси.

Може да забележите, че в конкретния пример и двете играят оптимално по този начин. Крайния резултат е 8 за Праси и 7 за Преси. Извеждаме 8.

**СУМИ**

Дадена ви е редица от N цели положителни числа. Тя трябва да бъде разделена от M разделителя на (М+1) части. Всяка от частите трябва да има сбор ненадвишаващ K. Намерете броя начини това да стане по модул MOD.

**Input Format**

* На първия ред от страндартния вход се въвеждат числата N, M, K и MOD
* На втория ред има N числа разделени с интервали - редицата

**Constraints**

* 1 ≤ N ≤ 100000
* 1 ≤ M ≤ 100
* 1 ≤ K ≤ 1018
* 1 ≤ MOD ≤ 109
* 1 ≤ числата от редицата ≤ 1018

**Output Format**

* На единствен ред на стандартния изход да се изведе търсения брой начини по модул MOD.

**Sample Input 0**

5 2 7 100

2 5 1 3 4

**Sample Output 0**

3

**Explanation 0**

Трите възможни разделяния са:

* 2 | 5 1 | 3 4
* 2 5 | 1 | 3 4
* 2 5 | 1 3 | 4