	<p>Metody programowania 2015/2016</p> <p>Maksymalna podtablica 2D o najmniejszej liczbie elementów</p>	<p>P_01</p>
---	--	-------------

Opis

Dla danej niepustej tablicy dwuwymiarowej liczb całkowitych: $a[0][0], \dots, a[n-1][m-1]$ dla $0 \leq i \leq j < n$, $0 \leq k \leq l < m$ definiujemy jej **maksymalną podtablicę** jako spójny fragment $a[i..j][k..l]$ o maksymalnej nieujemnej sumie elementów, obliczanej według wzoru: $s(i, j, k, l) = 3 \cdot D + 2 \cdot U$ gdzie:

D = suma dodatnich elementów $a[x][y]$ tej podtablicy, dla których $i \leq x \leq j$ oraz $k \leq y \leq l$.

U = suma ujemnych elementów $a[x][y]$ tej podtablicy, dla których $i \leq x \leq j$ oraz $k \leq y \leq l$.

W przypadku gdy wszystkie liczby tablicy są ujemne, maksymalna podtablica jest pusta i $s(i, j, k, l)$ jest równa 0. Natomiast gdy wszystkie liczby są równe 0 to maksymalna podtablica zawiera element $a[0][0]$ i też daje wynik 0.

Napisz w Javie program działający w czasie $O((\max(n, m))^3)$, który oblicza maksymalną wartość $s(i, j, k, l)$ oraz wyznacza maksymalną podtablicę $a[i..j][k..l]$ o najmniejszej liczbie elementów, której indeksy i, j, k, l tworzą ciąg leksykograficznie najmniejszy.

W przypadku, gdy maksymalna podtablica jest pusta wynik będzie równy liczbie 0. Natomiast gdy maksymalna podtablica istnieje wynik opisany będzie przez wartość $s(i, j, k, l)$ oraz cztery indeksy: i, j, k, l .

Wejście

Dane do programu wczytywane są ze standardowego wejścia (klawiatury) zgodnie z poniższą specyfikacją.

Pierwszą podawaną wartością będzie dodatnia liczba całkowita oznaczająca ilość zestawów danych, po której na wejściu pojawią się zestawy danych w ilości równej wczytanej liczbie.

Każdy zestaw danych zawiera dwie dodatnie liczby całkowite z zakresu od 1 do 100, oznaczające odpowiednio liczbę wierszy oraz liczbę kolumn tablicy, w następnych liniach podawane są dane będące kolejnymi wierszami tablicy zgodnie z podaną liczbą wierszy i kolumn.


Dane każdego zestawu są liczbami całkowitymi z zakresu od -2^{15} do $+2^{15}$.

Wyjście

Dla każdego zestawu danych jeśli maksymalna podtablica nie jest pusta program wypisuje dwie linie:

- w pierwszej linii : **max_sum=wartość $s(i, j, k, l)$**
- w drugiej linii: **$[i..j, k..l]$**
przy czym: i, j, k, l wyznaczają szukaną podtablicę **$a[i..j][k..l]$** ,
o najmniejszej liczbie elementów, której indeksy i, j, k, l tworzą ciąg leksykograficznie najmniejszy.

w przeciwnym przypadku program powinien wypisywać pojedynczy znak 0.

	<p>Metody programowania 2015/2016</p> <p>Maksymalna podtablica 2D o najmniejszej liczbie elementów</p>	<p>P_01</p>
---	--	-------------

Wymagania implementacyjne

1. W pierwszej linii program powinien zawierać komentarz:
// Imie Nazwisko – nr grupy
2. Jedynym dozwolonym importem jest obsługa wczytywania z klawiatury, to jest:
import java.util.Scanner;
3. Główna klasa musi nazywać się **Source**, co oznacza ogólne ramy kodu postaci:

```
class Source {
    public static void main( String [] args ) {
        ...
    }
}
```

4. Wczytywanie musi się odbywać przez pojedynczą zmienną skanera wczytywania, zadeklarowaną zewnętrznie w stosunku do wszystkich metod głównej klasy.

W praktyce oznacza to tylko jedną deklarację w przykładowej postaci:

```
public static Scanner scn = new Scanner(System.in);
```

w pierwszej linii ciała głównej klasy.

Przykład danych

wejście:	wyjście:
<pre>6 2 5 1 1 -1 -1 0 1 1 -1 -1 4 2 5 0 -1 -1 1 1 4 -2 -2 1 1 2 5 0 -1 -1 4 0 4 -2 -2 0 0 2 5 -1 -2 -3 -1 -2 -1 -1 -1 -1 -5 2 5 0 0 0 0 0 0 0 0 0 0 3 6 0 0 0 -2 1 1 0 1 1 -2 1 1 0 1 1 -2 0 0</pre>	<pre>max_sum=16 [0..1, 0..4] max_sum=12 [1..1, 0..0] max_sum=12 [0..0, 3..3] 0 max_sum=0 [0..0, 0..0] max_sum=12 [0..1, 4..5]</pre>