|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| http://ace.delos.com/usaco/cowhead2.gif | |  | | --- | | Contest: OPEN10 **GOLD** Division | |  | |  | |  | |

**ANALYSIS MODE  
Submit solutions for your own enjoyment.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**GOLD PROBLEMS**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Three problems numbered 1 through 3**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Problem 1: Cow Hopscotch [John Pardon, 2010]**

[**[](http://en.wikipedia.org/wiki/Hopscotch)**](http://en.wikipedia.org/wiki/Hopscotch)**The cows have reverted to their childhood and**

**are playing a game similar to human hopscotch.**

**Their hopscotch game features a line of N (3 <=**

**N <= 250,000) squares conveniently labeled 1..N**

**that are chalked onto the grass.**

**Like any good game, this version of hopscotch**

**has prizes! Square i is labeled with some**

**integer monetary value V\_i (-2,000,000,000 <=**

**V\_i <= 2,000,000,000). The cows play the game to**

**see who can earn the most money.**

**The rules are fairly simple:**

**\* A cow starts at square "0" (located just before square 1; it**

**has no monetary value).**

**\* She then executes a potentially empty sequence of jumps toward**

**square N. Each square she lands on can be a maximum of K (2**

**<= K <= N) squares from its predecessor square (i.e., from**

**square 1, she can jump outbound to squares 2 or 3 if K==2).**

**\* Whenever she wishes, the cow turns around and jumps back**

**towards square 0, stopping when she arrives there. In addition**

**to the restrictions above (including the K limit), two**

**additional restrictions apply:**

**\* She is not allowed to land on any square she touched on her**

**outbound trip (except square 0, of course).**

**\* Except for square 0, the squares she lands on during the**

**return trip must directly precede squares she landed on**

**during the outbound trip (though she might make some larger**

**leaps that skip potential return squares altogether).**

**She earns an amount of money equal to the sum of the monetary values**

**of all the squares she jumped on. Find the largest amount of cash**

**a cow can earn.**

**By way of example, consider this six-box cow-hopscotch course where**

**K has the value 3:**

**Square Num: 0 1 2 3 4 5 6**

**+---+ +---+ +---+ +---+ +---+ +---+ +---+**

**|///|--| |--| |--| |--| |--| |--| |**

**+---+ +---+ +---+ +---+ +---+ +---+ +---+**

**Value: - 0 1 2 -3 4 5**

**One (optimal) sequence Bessie could jump (shown with respective**

**bracketed monetary values) is: 1[0], 3[2], 6[5], 5[4], 2[1], 0[0]**

**would yield a monetary total of 0+2+5+4+1+0=12.**

**If Bessie jumped a sequence beginning with 0, 1, 2, 3, 4, ... then**

**she would be unable to return since she could not legally jump back**

**to an untouched square.**

**PROBLEM NAME: hop**

**INPUT FORMAT:**

**\* Line 1: Two space separated integers: N and K**

**\* Lines 2..N+1: Line i+1 contains a single integer: V\_i**

**SAMPLE INPUT (file hop.in):**

**6 3**

**0**

**1**

**2**

**-3**

**4**

**5**

**OUTPUT FORMAT:**

**\* Line 1: A single line with a single integer that is the maximum**

**amount of money a cow can earn**

**SAMPLE OUTPUT (file hop.out):**

**12**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Problem 2: Water Slides [John Pardon, 2010]**

**Inspired by the new water park at Machu**

**Picchu in Peru, Farmer John has decided**

**to build one for the cows. Its biggest**

**attraction is to be a giant water slide**

**of a peculiar design.**

**The superslide comprises E (1 <= E <=**

**150,000) mini slides connecting V (2 <=**

**V <= 50,000) small pools conveniently**

**labeled 1..V. Every mini slide must be**

**traversed in its proper direction and**

**may not be traversed backwards. The**

**cows start at pool number 1 and**

**traverse successive mini slides until**

**they end up in the pool number V, the**

**final pool. Every pool (except 1, the**

**first one) includes at least one mini**

**slide entering it and (except V, the**

**last one) at least one (different) mini**

**slide exiting it.**

**Furthermore, a cow can reach the**

**end of the ride (pool V) from any**

**pool by going down a sequence of mini**

**slides. Finally, since this is a**

**slide, it is not possible to leave a**

**pool and then encounter that pool**

**again after traversing some set of**

**mini slides.**

**Each mini slide i runs from pool P\_i to pool Q\_i (1 <= P\_i <= V; 1**

**<= Q\_i <= V; P\_i != Q\_i) and has an associated fun value F\_i (0 <=**

**F\_i <= 2,000,000,000). Bessie's total fun for any given trip down**

**the superslide is the sum of the fun values of all the mini slides**

**traversed.**

**Bessie naturally wants to have as much fun as possible, given the**

**long time that she spends in the slide's queue waiting for the ride.**

**Generally, she carefully chooses which mini slide to follow out of**

**each pool. She is a cow, however, and no more than K (1 <= K <= 10)**

**times as she splashes down the slide, she loses control and follows**

**a random mini slide out of a pool (this can even happen on pool 1).**

**If Bessie chooses so as to maximize her fun in the worst case, how**

**much fun is she guaranteed to have for a given super-slide?**

**By way of example, consider a small park that has 3 pools (pool**

**id's shown in brackets) and four mini slides; K has the value 1**

**(fun values shown outside of brackets):**

**[1]**

**/ \**

**5 -> / \ <- 9**

**/ \**

**[2]---3---[3]**

**\\_\_5\_\_/**

**She alway starts at pool 1 and ends and pool 3. If she had her way,**

**she'd ride direct from pool 1 to pool 2 and then on the higher-fun**

**mini slide (with fun value 5) to slide 3 for a total fun value of**

**5+5=10. But, if she loses control at pool 1, she might slide directly**

**from pool 1 to pool 3 for total fun 9. If she loses control at pool**

**2, she could reduce her total fun to just 5+3 = 8.**

**Bessie wants to find the most fun she can have so she strives to**

**choose 1->3 for a total fun of 9. If she loses control at pool 1**

**and ends up on mini slide 1->2, she knows she will not lose control**

**at pool 2 and will end up with fun 10. Thus, she knows her minimum**

**fun will always be at least 9.**

**PROBLEM NAME: slide**

**INPUT FORMAT:**

**\* Line 1: Three space separated integers: V, E, and K**

**\* Lines 2..E + 1: Line i+1 contains three space separated integers:**

**P\_i, Q\_i, and F\_i**

**SAMPLE INPUT (file slide.in):**

**3 4 1**

**2 3 5**

**1 2 5**

**1 3 9**

**2 3 3**

**OUTPUT FORMAT:**

**\* Line 1: A single line with a single integer that is the minimum fun**

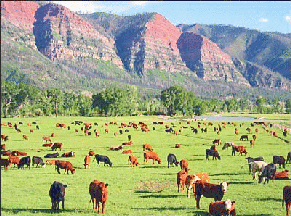
**that Bessie can guarantee she can have.**

**SAMPLE OUTPUT (file slide.out):**

**9**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Problem 3: Triangle Counting [Tom Conerly, 2010]**

**Bessie is standing guard duty after**

**the big bad wolf was spotted stalking**

**cows over at Farmer Don's spread.**

**Looking down from her guard tower in**

**utter boredom, she's decided to**

**perform intellectual exercises in**

**order to keep awake.**

**After imagining the field as an X,Y**

**grid, she recorded the coordinates of**

**the N (1 <= N <= 100,000)**

**conveniently numbered 1..N cows as**

**X\_i,Y\_i (-100,000 <= X\_i <= 100,000;**

**-100,000 <= Y\_i <= 100,000; 1 <= i <=**

**N). She then mentally formed all possible triangles that could be**

**made from subsets of the entire set of cow coordinates. She counts**

**a triangle as 'golden' if it wholly contains the origin (0,0). The**

**origin does not fall on the line between any pair of cows. Additionally,**

**no cow is standing exactly on the origin.**

**Given the list of cow locations, calculate the number of 'golden'**

**triangles that contain the origin so Bessie will know if she's doing**

**a good job.**

**By way of example, consider 5 cows at these locations:**

**-5,0 0,2 11,2 -11,-6 11,-5**

**Below is a schematic layout of the field from Betsy's point of view:**

**............|............**

**............\*..........\*.**

**............|............**

**-------\*----+------------**

**............|............**

**............|............**

**............|............**

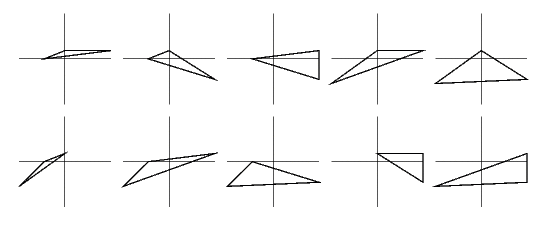
**............|............**

**............|..........\*.**

**.\*..........|............**

**............|............**

**All ten triangles below can be formed from the five points above:**

****

**By inspection, 5 of them contain the origin and hence are 'golden'.**

**PROBLEM NAME: tricount**

**INPUT FORMAT:**

**\* Line 1: A single integer: N**

**\* Lines 2..N+1: Each line contains two integers, the coordinates of a**

**single cow: X\_i and Y\_i**

**SAMPLE INPUT (file tricount.in):**

**5**

**-5 0**

**0 2**

**11 2**

**-11 -6**

**11 -5**

**OUTPUT FORMAT:**

**\* Line 1: A single line with a single integer that is the count of the**

**number of times a triangle formed by the cow locations**

**contains the origin**

**SAMPLE OUTPUT (file tricount.out):**

**5**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***