****
****
GOLD PROBLEMS
****
Three problems numbered 1 through 3
****

Problem 1: Ski Lift [Adam Rosenfield, 2004]

Farmer Ron in Colorado is building a ski resort for his cows (though budget constraints dictate construction of just one ski lift). The lift will be constructed as a monorail and will connect a concrete support at the starting location to the support at the ending location via some number of intermediate supports, each of height 0 above its land. A straight-line segment of steel connects every pair of adjacent supports. For obvious reasons, each section of straight steel must lie above the ground at all points.

Always frugal, FR wants to minimize the number of supports that he must build. He has surveyed the N (2 <= N <= 5,000) equal-sized plots of land the lift will traverse and recorded the integral height H (0 <= H <= 1,000,000,000) of each plot. Safety regulations require FR to build adjacent supports no more than K (1 <= K <= N - 1) units apart. The steel between each pair of supports is rigid and forms a straight line from one support to the next.

Help FR compute the smallest number of supports required such that: each segment of steel lies entirely above (or just tangent to) each piece of ground, no two consecutive supports are more than K units apart horizontally, and a support resides both on the first plot of land and on the last plot of land.

PROBLEM NAME: skilift

INPUT FORMAT:

* Line 1: Two space-separate integers, N and K

* Lines 2..N+1: Line i+1 contains a single integer that is the height of plot i.

SAMPLE INPUT (file skilift.in):

```
13 4
0
1
0
2
4
6
8
6
8
8
9
11
12
```

OUTPUT FORMAT:

* Line 1: A single integer equal to the fewest number of lift towers
  FR needs to build subject to the above constraints

SAMPLE OUTPUT (file skilift.out):

5

OUTPUT DETAILS:

FR builds five supports (at locations 1, 5, 7, 9, and 13). The steel
is tangent to the ground at four locations: 1-5, 5-7, 7-9, and
12-13.

If FR only builds supports at the four locations 1, 5, 9, and 13,
then the steel would be below ground from 5-9. If FR built supports
at 1, 7, and 13, then -- although the steel is always above ground
-- supports 7 and 13 are more than 4 units apart horizontally. There
is no solution using fewer than 5 supports such that no two consecutive
supports are more than 4 units apart horizontally.

****

Problem 2: Milk Team Select [Percy Liang, 2002]

Farmer John's N (1 <= N <= 500) cows are trying to select the
milking team for the world-famous Multistate Milking Match-up (MMM)
competition. As you probably know, any team that produces at least
X (1 <= X <= 1,000,000) gallons of milk is a winner.

Each cow has the potential of contributing between -10,000 and
10,000 gallons of milk. (Sadly, some cows have a tendency to knock
over jugs containing milk produced by other cows.)

The MMM prides itself on promoting family values. FJ's cows have
no doubt that they can produce X gallons of milk and win the contest,
but to support the contest's spirit, they want to send a team with
as many parent-child relationships as possible (while still producing
at least X gallons of milk). Not surprisingly, all the cows on FJ's
farm are female.

Given the family tree of FJ's cows and the amount of milk that each
would contribute, compute the maximum number of parent-child
relationships that can exist in a winning team. Note that a set of
cows with a grandmother-mother-daughter combination has two
parent-child relationships (grandmother-mother, mother-daughter).

PROBLEM NAME: tselect

INPUT FORMAT:

* Line 1: Two space-separated integers, N and X.

* Lines 2..N+1: Line i+1 contains two space-separated integers
  describing cow i. The first integer is the number of gallons

of milk cow i would contribute. The second integer (range 1..N) is the index of the cow's mother. If the cow's mother is unknown, the second number is 0. The family information has no cycles: no cow is her own mother, grandmother, etc.

SAMPLE INPUT (file tselect.in):

```
5 8
-1 0
3 1
5 1
-3 3
2 0
```

INPUT DETAILS:

There are 5 cows. Cow 1 can produce -1 gallons and has two daughters, cow 2 and 3, who can produce 3 and 5 gallons, respectively. Cow 3 has a daughter (cow 4) who can produce -3 gallons. Then there's cow 5, who can produce 2 gallons.

OUTPUT FORMAT:

* Line 1: The maximum number of parent-child relationships possible on a winning team. Print -1 if no team can win.

SAMPLE OUTPUT (file tselect.out):

```
2
```

OUTPUT DETAILS:

The best team consists of cows 1, 2, 3, and 5. Together they produce (-1)+3+5+2 = 9 >= 8 gallons and have 2 parent-child relationships (1--2 and 1--3). Note that a team with cows 2, 3, and 5 would be able to produce more milk (10 gallons), but would have fewer parent-child relationships (0).

****

Problem 3: Mooo [Brian Dean, 2005]

Farmer John's N (1 <= N <= 50,000) cows are standing in a very straight row and mooing. Each cow has a unique height h in the range 1..2,000,000,000 nanometers (FJ really is a stickler for precision). Each cow moos at some volume v in the range 1..10,000. This "moo" travels across the row of cows in both directions (except for the end cows, obviously). Curiously, it is heard only by the closest cow in each direction whose height is strictly larger than that of the mooing cow (so each moo will be heard by 0, 1 or 2 other cows, depending on whether or not taller cows exist to the mooing cow's right or left).

The total moo volume heard by given cow is the sum of all the moo volumes v for all cows whose mooing reaches the cow. Since some (presumably taller) cows might be subjected to a very large moo volume, FJ wants to buy earmuffs for the cow whose hearing is most threatened. Please compute the loudest moo volume heard by any cow.

PROBLEM NAME: mooo

INPUT FORMAT:

* Line 1: A single integer, N.

* Lines 2..N+1: Line i+1 contains two space-separated integers, h and
v, for the cow standing at location i.

SAMPLE INPUT (file mooo.in):

3
4 2
3 5
6 10

INPUT DETAILS:

Three cows: the first one has height 4 and moos with volume 2, etc.

OUTPUT FORMAT:

* Line 1: The loudest moo volume heard by any single cow.

SAMPLE OUTPUT (file mooo.out):

7

OUTPUT DETAILS:

The third cow hears both the first and second cows moo 2+5=7. Though the third cow moos with volume 10, no one hears her.

****
****
SILVER PROBLEMS
****
Three problems numbered 6 through 8
****

Problem 6: Mooo [Brian Dean, 2005]

Farmer John's N (1 <= N <= 50,000) cows are standing in a very straight row and mooing. Each cow has a unique height h in the range 1..2,000,000,000 nanometers (FJ really is a stickler for precision). Each cow moos at some volume v in the range 1..10,000. This "moo" travels across the row of cows in both directions (except for the end cows, obviously). Curiously, it is heard only by the closest cow in each direction whose height is strictly larger than that of the mooing cow (so each moo will be heard by 0, 1 or 2 other cows, depending on not whether or taller cows exist to the mooing cow's right or left).

The total moo volume heard by given cow is the sum of all the moo volumes v for all cows whose mooing reaches the cow. Since some

(presumably taller) cows might be subjected to a very large moo
volume, FJ wants to buy earmuffs for the cow whose hearing is most
threatened. Please compute the loudest moo volume heard by any cow.

PROBLEM NAME: moooo

INPUT FORMAT:

* Line 1: A single integer, N.

* Lines 2..N+1: Line i+1 contains two space-separated integers, h and
v, for the cow standing at location i.

SAMPLE INPUT (file moooo.in):

3
4 2
3 5
6 10

INPUT DETAILS:

Three cows: the first one has height 4 and moos with volume 2, etc.

OUTPUT FORMAT:

* Line 1: The loudest moo volume heard by any single cow.

SAMPLE OUTPUT (file moooo.out):

7

OUTPUT DETAILS:

The third cow hears both the first and second cows moo 2+5=7. Though the
third cow moos with volume 10, no one hears her.

****

Problem 7: Water Slides [Eric Price, 2006]

It's a hot summer day, and Farmer John is letting Betsy go to the
water park where she intends to ride every single slide. The water
park has N (1 <= N <= 10,000) platforms (numbered 1..N) from which
to enter the M (1 <= M <= 10,000) water slides. Each water slide
starts at the top of some platform and ends at the bottom of some
platform (possibly the same one). Some platforms might have more
than one slide; some might not have any.

The park is very thin, so the platforms lie along a straight line,
each platform at a position Xi (0 <= Xi <= 100,000) meters from one
end of the park. One walks from one platform to the next via a
sidewalk parallel to the line of platforms.

The platforms of the water park are weakly connected; that is, the
park cannot be divided into two sets of platforms with no slides
running between the two sets. Both the entrance and exit to the

park are at platform 1, so Betsy will start and end there.

In order to spend more time on the slides, Betsy wants to walk as little as possible. Find the minimum distance Betsy must travel along the ground in order to try every slide in the park exactly once without repeating.

PROBLEM NAME: slides

INPUT FORMAT:

* Line 1: Two integers, N and M.

* Lines 2..N+1: Line i+1 contains one integer, Xi, the position of platform i.

* Lines N+2..M+N+1: Line i+N+1 contains two integers, Si and Di, respectively the start and end platforms of a slide.

SAMPLE INPUT (file slides.in):

```
5 7
5
3
1
7
10
1 2
1 2
2 3
3 1
4 5
1 5
4 1
```

OUTPUT FORMAT:

* Line 1: One integer, the minimum number of meters Betsy must walk.

SAMPLE OUTPUT (file slides.out):

```
8
```

OUTPUT DETAILS:

Betsy slides from 1 -> 2 -> 3 -> 1 -> 2, then walks a distance 2 back to platform 1. She can then slide to 5, walk to 4, slide to 5, walk back to 4 and slide to 1 for a total distance of 8.

****

Problem 8: Lights Out [Rob Kolstad, 2005]

The cows so much prefer to sleep in the dark. Every night, though, when they return to the barn, some of the L (3 <= L <= 50) lights are switched on. They know the location of the push button switches but, as is always the problem, they lack fingers. The buttons are arranged in a nice row, and the left most button (#1) toggles light #1; the next button to the right toggles light #2; and so on. ('Toggle' means change from off-to-on or on-to-off, depending on the current state of the switch.)

They do however have an unusual pitchfork with T (1 <= T <= 7) slots, each one of which might hold a tine that can push the switches. Some tines are present; some might be absent. By way of example, imagine a pitchfork with T=4 and a missing tine. This particular pitchfork has tines in the 1st, 2nd, and 4th positions, easily described as '1101'.

If the pitchfork is aimed at the leftmost switch, then lights #1, #2, and #4 are toggled (#3 isn't toggled since there is no tine there). If the pitchfork is aimed at switch #3, then lights #3, #4, and #6 are toggled. The pitchfork must be aimed so that all its tine slots touch switches; the fork cannot cross the end of the line of switches.

Given a list of lights that are on and a configuration for a pitchfork, determine a sequence of pitchfork presses that will toggle the lights until the minimum number of lights is lit.

PROBLEM NAME: xlite

INPUT FORMAT:

* Line 1: Two space-separated integers: L and T

* Line 2: A line with L characters (and no spaces), each of which is '0' or '1'. '1' means a light in that slot is lit; '0' means the light in that slot is not lit.

* Line 3: A line with T characters, each of which is '0' or '1' (no spaces are present). '1' means a tine is present on the pitchfork in that slot; '0' means otherwise. The pitchfork can not be inverted.

SAMPLE INPUT (file xlite.in):

```
10 4
1111111111
1101
```

INPUT DETAILS:

All 10 lights all on; three of four tines are present on the pitchfork (tine #3 is missing)

OUTPUT FORMAT:

* Line 1: K, the number of positions at which the pitchfork was aimed.

* Lines 2..K+1: Each line contains a single integer X (1 <= X <= L-T+1) that tells where the pitchfork was aimed (the leftmost tine slot, that is) to toggle some lights. A program will evaluate the decisions to simulate toggling the lights. Full

credit is given for minimal-length solutions that leave to the minimum number of lights turned on; partial credit for solutions that take longer to minimize the lights. It is not always possible switch all the lights off.

SAMPLE OUTPUT (file xlite.out):

5
3
1
4
7
6

OUTPUT DETAILS:

1111111111   Start
1100101111   Toggle 3
0001101111   Toggle 1
0000000111   Toggle 4
0000001010   Toggle 7
0000010000   Toggle 6
One light remaining is the best one can do.   Many other solutions will leave one light lit (possibly a different light).

****