Editorial: Horses

To solve 1st subtask we need just calculate our answer using dynamic programming dp[i][j] - what is maximal profit if we pass i-1 days and we have j horses then we got $j * x_i$ horses and check all number of horses that we sell today and go to the next day.

#1 Observation

First of all let’s consider to points i and j, what if we sell k horses in i-th and j-th day. And check in what day it's more preferable.

Profit of i-th day: $x_1 * x_2 * ... * x_i * y_i$

Profit of j-th day: $x_1 * x_2 * ... * x_i * x_{i+1} * ... * x_j * y_j$

It depends on this

y_i ? $x_{i+1} * ... * x_j * y_j$

>

<

=

so if it's > then it's profitably sell horses in i-th day

if it's < then it's profitably sell horses in j-th day

if it's = then it's no matter when we sell them in i-th day or in j-th day

From this point it's clear that it is better to sell all our horses in one day that gives us maximal profit.

Using this observation we can solve 2 subtasks.

After each query we can solve problem in O(n)

#2 Observation

If all $x_i >= 2$, then after 30 days $x_i * x_{i+1} * x_{i+2} * ... * x_{i+29} >= 2^{30} > 10^9$, so position less then and equal n-30 never can be optimal solution, because even if $y_{n-30} = 10^9$ and $y_n = 1$, $2^{30} * y_n > y_{n-30}$

It's enough to check only last 30 positions

#3 Observation

The main problem in last subtask is $x_i = 1$, but in that cases multiplication first x_i numbers and x_{i-1} is not change, this gives us opportunity to merge consecutive positions with $x_i = 1$, when we merge this positions we should take the maximal y_i. So if we do that, it's enough to check last 60 positions, because it can be no more than 30 merged 1's between 30 last positions where $x_i >= 2$.

So we can store our state in some structure like set, that provide us information about current merged positions, and some structure that provide us RMQ. So solution per query will be $\log(10^9) * \log(n)$. Then we have to calculate answer we can use something like segment tree.

So overall complexity will be $O(m * \log(10^9) * \log(n))$.