A picture containing text, tiled, net, tile

Description automatically generated



**4)** We have array with N\*N, so ups and downs could be approximated to N

\*-1 if it goes higher;

if(a[i][j]<a[i+L][j+H]) then

energy--;

\*+1 if it goes lower;

if(a[i][j]>a[i+L][j+H]) then

energy++;

\*+0 if it moves horizontally.

if(a[i][j]==a[i+L][j+H]) then

//does nothing

Solution for case on image: 0 -1(/) +6(\) -6(/) +10(\) -7(/) +**12(\)** -11(/) +5(\) -11(/) +1(\) => -2

Energy value for each peak: 0; -1; +5; -1; +9 ; +2; +14; +3; +8; -3; -2.

Here it is clear that **goat gains the highest energy on decline after third peak.**

**6)** We have array with N\*N, so ups and downs could be approximated to N

\*For each step, the camel spends 1 unit of water;

water--;

\*if at the same time it rises higher, then it spends another 1 unit of water (two times more water);

if(a[i][j]<a[i+L][j+H]);

water -=2;

\*At the points of the local minimum, it is filled with 10 units of water;

isMinimum = false

if(a[i][j]<a[i+L][j+H]) then

isMinimum = true;

water += 10;

Solution for case on image: 0 -2(/)-6(\)+10(min)-12(/)-10(\)+10(min)-14(/)-24(\)+10(min)-22(/)-5(\)+10(min)-22(/)-1(\)+10(min) =>-68

Water value for each peak: 0; -2; 2; -10; -10; -24; -38; -60; -55; -77; -68; The most beneficial is **last decline**.