0 based indexing

Problem A:
We will follow following recursion : ans[i]=a[i]+max(ans[i+1],0);
Base case : ans[n-1]=a[n-1]
Complexity : O(n)

Problem B:
Solution 1:
Binary search on maximum height of buildings, Check if its current height can be achieved with cost less than or equal to K.
Complexity : O(nlog(max(a_i)));
Solution 2:
Sort all the buildings in increasing order of height:

For all heights h : a[i]<=h<a[i+1]; Check the least h such that cost encountered is less than equal to K.
For each i it will take O(1) operations if done smartly.
O(nlog(n))

Problem C:
Recursion → a[i]=a[i-1]+a[i-2]+a[i-3]...a[i-k];
We can transform a vector this way.
Vector A ={a[i], a[i-1], a[i-2], a[-3]..a[i-k+1]}
Vector B= {a[i+1], a[i], a[i-1], a[i-2], a[-3]..a[i-k+2]}
The matrix for M of size k*k be.
1 1 1 1 1 …1
1 0 0 0 0… 0
0 1 0 0 0… 0
0 0 1 0 0.. 0
0 0 0 1 0 … 0
.
.
.
.
0 0 0 0….. 1
Then MA=B
Now we can use matrix exponentiation and solve the problem.
Complexity O(k^3log(n))