Answer : 2(a)

Name: Shabab Abdullah

ID: 20301005

Implementation 01:

$T(n) = T(n-1) + T(n-2) + C$

$= 2T(n-2) + C$

Here,
$T(n-1) \approx T(n-2)$

$= 2(2T(n-2) + c) + C$

$= 4T(n-2) + 3C$

$= 8T(n-3) + 7C$

$T(n) = 2^k T(n-k) + (2^k - 1)C$

if we assume, $n-k=0$

$\Rightarrow n = k$

So, $T(n) = 2^n C - C$

$= 2^n$

$\therefore$ The time complexity will be $O(2^n)$

Implementation 02

$T(n) = n-2$

So, $T(n) = n$

So, time complexity will be $O(n)$

Now, here we can see the first implementation is $O(2^n)$ and the second implementation is $O(n)$. So we can say the second implementation will take sorter period of time than the first one. So, the second implementation is much faster than the first implementation.