2. Implementation
$$-1$$

def bibonacci_1(n):

if $n < 0$:

print('Invalid input')

elif $n < 2$:

return $n - 1$

else:

return fibonacci_1(n-1)+fibonacci_1(n-2)

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

$$= 2T(n-1)+1 \left[T(n-1) \approx T(n-2)\right]$$

$$= 2 \left\{2T(n-2)+1\right\}+1$$

$$= 2^2 T(n-2)+2^1+2^0$$

$$= 2^2 \left\{2T(n-3)+1\right\}+2^2+2^0$$

$$= 2^2 T(n-3)+2^2+2^1+2^0$$

$$= 2^{n+2} T(n-n+2)+2^{n+1}+\dots 2^0$$

$$= 2^{n+2} 1+2^{n+1}+\dots 2^0$$

$$= 2^{n+2} - 1$$

$$= 2^{n+2} - 1$$

$$= 0(2^n)$$

The state of the s	
bibonacci_annay = $[0,1]$ } $0(1)$ if $n < 0$: praint ('Invalid input!")} $0(1)$ elif $n < = 2$: rectain bibonacci_annay $[n-1]$ } $0(1)$ else: bon i in range $(2,n)$: $[1-1]$ + bibonacci_annay $[1-1]$ + bibonacci_annay $[1-1]$ + bibonacci_annay $[1-1]$ + bibonacci_annay Time complexity = $0(1) + 0(n)$	Implementation -2
elif n<=2: rectain bibonacci-annay [n-1] } O(1) else: bon i in range (2,n): [i-i]+bibonacci-annay [i-2] = O(n) noturn bibonacci-annay [-1] Time complexity = O(1) + O(n)	def fibonacci_2(n):
elif n<=2: rectain bibonacci-annay [n-1] } O(1) else: bon i in range (2,n): [i-i]+bibonacci-annay [i-2] = O(n) noturn bibonacci-annay [-1] Time complexity = O(1) + O(n)	fibonacci_armay = [0,1] } O(1)
elif $n \le 2$: redurn bibonacci-anray $[n-1] \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$	if n<0: proint ('Invalid input!")}0(1)
else: ben i in range (2,n): $O(1) + n$ fibonacci_annay.append (fibonacci_annay [i-2]) $= O(n)$ roturn fibonacci_annay [-1] Time complexity = $O(1) + O(n)$	
O(1) + n $= O(n)$ $= O(n)$ $= O(n)$ $= O(n)$ $= O(n)$ $= O(n)$ $= O(n) + 0(n)$ $= O(n) + 0(n)$ $= O(n) + O(n)$	else:
= $O(n)$ reduce bibonacci-annay [-1] Time complexity = $O(1) + O(n)$	for i in range (2,n):
= $O(n)$ reduce bibonacci-annay [-1] Time complexity = $O(1) + O(n)$	(1) x-n { filoonacci_annay.append (filoonacci_annay[i-2]
Time complexity = $O(1) + O(1)$	= O(n) noturn fibonacci-annay [-]
	Time complexity = $O(1) + O(1)$

\$. How (1 (10) - (10) and 1 (10) 4. For i=0 to n-1for j=0 to n-1for k=0 to n-1O(n)

O(n)

C[i,j] t=A[i,k]*B[k,j]end for

end for

end for Time complexity = O(n(n(n)))= 0(n3)