



UAEM

Universidad Autónoma
del Estado de México



Universidad Autónoma del Estado de México

Centro Universitario UAEM Atlacomulco

“Problemas de recursividad”

Presenta:

Elias Edgardo Segundo Antonio

Revisa:

Ingeniero José Luis García Morales

Unidad de Aprendizaje:

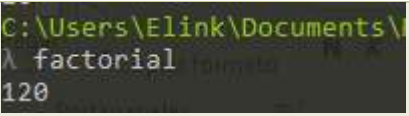
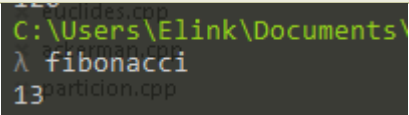
Programación avanzada

Fecha de entrega:


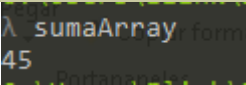
15 de mayo de 2017



A continuación, se muestran los ejercicios realizados de la sección 4.n del libro estructuras de datos 3ra edición, se muestra el código y una captura de pantalla de la ejecución.

Nombre	Código	Ejecución
factorial	<pre>#include <iostream> using namespace std; long long int factorial(int n) { if(n==0 n==1) { return 1; } else return n*factorial(n-1); } int main() { cout<<factorial(5); }</pre>	
Fibonacci	<pre>#include <iostream> using namespace std; long long int fibonacci(int n) { if (n==0 or n==1) { return n; } return fibonacci(n-1)+fibonacci(n-2); } int main() { cout<<fibonacci(7); }</pre>	



Imprimir arreglo	<pre>#include <iostream> #include <vector> using namespace std; void imprimirArray(vector<int> A,int i) { if(i<A.size()) { cout<<A[i]<<"\t"; imprimirArray(A,i+1); } } int main() { vector <int> A; for(int i=0;i<10;i++) A.push_back(i); imprimirArray(A,0); }</pre>	
Suma de un arreglo	<pre>#include <iostream> #include <vector> using namespace std; int sumaArray(vector<int> A,int i) { if(i<A.size()) { return A[i]+sumaArray(A,i+1); } } int main() { vector <int> A; for(int i=0;i<10;i++) A.push_back(i); cout<<sumaArray(A,0); }</pre>	



Euclides	<pre>#include <iostream> using namespace std; float mcd(int m, int n) { if (n==0) { return m; } else return mcd(n,m%n); } int main() { cout<<mcd(12,40); }</pre>	<pre>C:\Users\Elink\Documents\ λ euclides 4 Recuperación de doc</pre>
Ackermann	<pre>#include <iostream> using namespace std; int ackermann(int m, int n) { if(m==0) return n+1; if(m>0 && n==0) return (ackermann(m-1,1)); if(m>0 && n>0) return (ackermann(m-1, ackermann(m,n-1))); } int main() { cout<<ackermann(3,0); }</pre>	<pre>C:\Users\Elink\Documents\ λ ackerman 5</pre>
Particion	<pre>#include <iostream> using namespace std; int particion(int m,int n) { if(m==1 or n==1) return 1; if (m<n)return particion(m,m); }</pre>	<pre>C:\Users\Elink\Documents\ λ particion 6 portada [Recuperado aut...</pre>



	<pre> if (m==n) return 1+particion(m-1,m); if (m>n) return particion(m,n- 1)+particion(m-n,n); } int main() { cout<<particion(6,6); } </pre>	
catalán	<pre> #include <iostream> using namespace std; long long int factorial(int n) { if(n==0 or n==1) { return n; } else return n*factorial(n-1); } long long int catalan(int n) { return (factorial(2*n)) / (factorial(n)*factorial(n+1)); } int main() { cout<<catalan(5); } </pre>	<pre> C:\Users\Elink\Documents> λ catalan 42 </pre>
Coeficientes binomiales	<pre> #include <iostream> using namespace std; int CB(int n, int k) { if (k==0 or n==k) </pre>	<pre> C:\Users\Elink\Documents> λ cb 10 </pre>



	<pre>{ return 1; } else{ return CB(n-1,k- 1)+CB(n-1,k); } } int main() { cout<<CB(5,2); }</pre>	
--	--	--