



Bienvenido(a),
Visitante. Por favor
Ingresar o [Registrarse](#)
¿Perdiste tu email de
activación?.

| [Foro](#) | [Web](#) | [Blog](#) | [Wiki](#) | [Ayuda](#) | [Buscar](#) | [Ingresar](#) | [Registrarse](#) | 17 Octubre 2017, 02:12

Tema destacado: [AIO elhacker.NET 2015] **Compilación herramientas análisis y desinfección malware**

Foro de elhacker.net
 Programación
 Programación General
 Ejercicios
 Ejercicios Recursivos en Java y sus Soluciones

0 Usuarios y 6 Visitantes están viendo este tema.

Páginas: [1] 2 3 4 5 [ir abajo](#)

[responder](#)

[imprimir](#)

Autor

Tema: Ejercicios Recursivos en Java y sus Soluciones (Leído 336,299 veces)

@ohk



Ejercicios Recursivos en Java y sus Soluciones

« en: 11 Octubre 2008, 03:19 »



Desconectado

Mensajes:
1.728



Yo lo que
quiero que me
salga bien es
la vida.



Planteamiento del Ejercicio acompañado del algoritmo de resolución en Java

Planteamiento:

Ejercicio 1. Programar un algoritmo recursivo que calcule el factorial de un número.

Solución:

Código

```
1.     int factorial(int n){
2.         if(n==0) return 1;    //AXIOMA
3.         else return n*factorial(n-1); //FORMULA RECURSIVA
4.     }
```

Planteamiento:

Ejercicio 2. Programar un algoritmo recursivo que calcule un número de la serie fibonacci.

Solución:

Código

```
1.     int fibonacci(int n){
2.         if(n==1 || n==2) return 1;
3.         else return fibonacci(n-1)+fibonacci(n-2);
4.     }
```

Planteamiento:

Ejercicio 3. Programar un algoritmo recursivo que permita hacer la división por restas sucesivas.

Solución:

Código

```
1.    int division (int a, int b)
2.    {
3.        if(b > a) return 0;
4.        else
5.            return division(a-b, b) + 1;
6.    }
7.
```

Planteamiento:

Ejercicio 4. Programar un algoritmo recursivo que permita invertir un número.

Ejemplo: Entrada: 123 Salida: 321

Solución:

Código

```
1.    int invertir (int n)
2.    {
3.        if (n < 10)           //caso base
4.            return n;
5.        else
6.            return (n % 10) + invertir (n / 10) * 10;
7.    }
```

Planteamiento:

Ejercicio 5. Programar un algoritmo recursivo que permita sumar los dígitos de un número. **Ejemplo:** Entrada: 123 Resultado:6

Solución:

Código

```
1.    int sumar_dig (int n)
2.    {
3.        if (n == 0)           //caso base
4.            return n;
5.        else
6.            return sumar_dig (n / 10) + (n % 10);
7.    }
```

Planteamiento:

Ejercicio 6. Programar un algoritmo recursivo que permita hacer una multiplicación, utilizando el método Ruso. Para mas informacion: [aqui](http://foro.elhacker.net/ejercicios/ejercicios_recursivos_en_java_y_sus_soluciones-t231013.0.html).

Solución:

Código

```
1.    int mult_rusa(int A, int B)
2.    {
```

```
3.         if(A==1){
4.             return (B);
5.         }
6.         if(A%2!=0){
7.             return(B+mult_rusa( A/2 , B*2));
8.         }
9.         else{
10.            return(mult_rusa( A/2 , B*2));
11.        }
12.    }
```

Planteamiento:

Ejercicio 7. Programar un algoritmo recursivo que permita sumar los elementos de un vector.

Solución:

Código

```
1.     int suma_vec(int v [], int n)
2.     {
3.         if (n == 0)
4.             return v [n];
5.         else
6.             return suma_vec(v, n - 1) + v [n];
7.     }
```

Planteamiento:

Ejercicio 8. Programar un algoritmo recursivo que permita multiplicar los elementos de un vector.

Solución:

Código

```
1.     int multiplicar (int vec [], int tam)
2.     {
3.         if (tam == 0)
4.             return (vec [0]);
5.         return (vec [tam] * multiplicar (vec, tam - 1));
6.     }
```

Planteamiento:

Ejercicio 9. Programar un algoritmo recursivo que calcule el Maximo comun divisor de dos números.

Solución:

Código

```
1.     int sacar_mcd(int a, int b) {
2.         if(b==0)
3.             return a;
4.         else
5.             return sacar_mcd(b, a % b);
6.     }
7.     }
```

Planteamiento:

Ejercicio 10. Programar un algoritmo recursivo que determine si un número es positivo.

Solución:

Código

```
1. public boolean positivo(int n){
2.     if(n>0) return true;
3.     else return negativo(n);
4. }
5.
6. public boolean negativo(int n){
7.     if(n<0) return false;
8.     else return positivo(n);
9. }
```

Planteamiento:

Ejercicio 11. Programar un algoritmo recursivo que determine si un número es impar utilizando recursividad cruzada.

Solución:

Código

```
1. public boolean par(int n){
2.     if(n==0) return true;
3.     else return impar(n-1);
4. }
5.
6. public boolean impar(int n){
7.     if(n==0) return false;
8.     else return par(n-1);
9. }
```

Planteamiento:

Ejercicio 12. Programar un algoritmo recursivo que permita sumar los elementos de una matriz.

Solución:

Código

```
1. int suma (int fila, int col, int orden, int mat [] [])
2. {
3.     if (fila == 0 && col == 0)
4.         return mat [0] [0];
5.     else
6.         if (col < 0)
7.             return suma (fila - 1, orden, orden, mat);
8.         else
9.             return mat [fila] [col] + suma (fila, col - 1, or
10. }
```

Planteamiento:

Ejercicio 13. Programar un algoritmo recursivo que permita resolver el cuadro latino. **Ejemplo de cuadro latino:**

0 0 0 0 1

```

0 0 0 1 2
0 0 1 2 3
0 1 2 3 4
1 2 3 4 5

```

Solución:**Código**

```

1.     latino (int fila, int col, int cont, int orden, int mat [] [
2.     {
3.         if (fila == 0 && col == 0)
4.             mat [0] [0] = 1;
5.         else
6.             if (fila == col)
7.                 latino (fila - 1, orden - 1, orden, orden, mat);
8.             else
9.                 {
10.                    mat [fila] [col] = cont;
11.                    latino (fila, col - 1, orden + 1, orden, mat);
12.                }
13.     }

```

Planteamiento:

Ejercicio 14. Programar un algoritmo recursivo que permita resolver la siguiente matriz:

```

1 1 1 1 1
1 2 2 2 2
1 2 4 4 4
1 2 4 8 8
1 2 4 8 16

```

Solución: Solucionado por: **AmeRiK@nO**

Código

```

1. public class MatrizRecursividad {
2.
3.     private static int a=0, aux=1, b=0; //Declaramos Los dato
4.     private static int[][] matriz = new int[6][6]; //La matri
5.
6.     public static void main(String[] args) {
7.
8.         llenarMatriz(matriz, a, b); //Iniciamos el llamad
9.         imprimir(); //imprimimos la matriz
10.
11.     }
12.
13.     public static void llenarMatriz(int matriz[][], int i, in
14.
15.         if(j > matriz.length -1){ //Si Llegó a la ultima
16.             i++;
17.             j=0;
18.             aux++;
19.         }
20.         if(i <matriz.length){ // compara que no hallamos
21.
22.             if(i==(aux-1) && j >= (aux-1)){ //comprue
23.                 if(i==0)// si es la primera fila
24.                     matriz[i][j] = matriz[i][
25.                 else
26.                     matriz[i][j] = matriz[i][
27.                     llenarMatriz(matriz, i , j+1);
28.             }
29.             else{ //si no, asignamos Los valores ante
30.                 if(j==0)// comprobamos si es el p
31.                     matriz[i][j] = j+1;

```

```

32.         else
33.             matriz[i][j] = matriz[i-1
34.             llenarMatriz(matriz, i, j+1);
35.         }
36.     }
37. }
38.
39. public static void imprimir(){ //este metodo nos imprime
40.
41.     for(int i=0; i< matriz.length; i++){
42.         for(int j=0; j< matriz.length; j++){
43.             System.out.print(matriz[i][j] + "
44.         }
45.         System.out.print("\n");
46.     }
47. }
48.
49. }

```

Planteamiento:

Ejercicio 15. Programar un algoritmo recursivo que ejecute la matriz del cubo mágico.

Solución:**Código**

```

1. void magico(int mat [] [], int fil, int colmedio, int c, int
2. {
3.     if (c == n * n)
4.     {
5.         mat [n-1] [colmedio] = c;
6.     }
7.     else
8.     {
9.         if (fil < 0 && colmedio == n)
10.        {
11.            magico(mat, fil + 2, n - 1, c, n);
12.        }
13.        else
14.        {
15.            if (fil < 0)
16.            {
17.                magico(mat, n - 1, colmedio, c, n);
18.            }
19.            else
20.            {
21.                if (colmedio == n)
22.                {
23.                    magico(mat, fil, 0, c, n);
24.                }
25.                else
26.                {
27.                    if (mat [fil] [colmedio] == 0)
28.                    {
29.                        mat [fil] [colmedio] = c;
30.                        magico(mat, fil - 1, colmedio + 1, c
31.                    }
32.                    else
33.                    {
34.                        magico(mat, fil + 2, colmedio - 1, c,
35.                    }
36.                }
37.            }
38.        }
39.    }
40. }
41. }

```

Planteamiento:

Ejercicio 16. Programar un algoritmo recursivo que muestre el numero menor de un vector.

Solución:**Código**

```
1.      int menorvec (int x [], int n, int menor) {
2.      if (n == 0)
3.          if (menor > x [n]) return x [0];
4.          else return menor;
5.      else
6.          if (menor > x [n]) return menorvec (x, n - 1, x [n]);
7.          else return menorvec (x, n - 1, menor); }
```

Planteamiento:

Ejercicio 17. Programar un algoritmo recursivo que muestre el numero mayor de un vector.

Solución:**Código**

```
1.      int mayor (int numeros [], int posicion) {
2.      int aux;
3.      if (posicion == 0) return numeros [posicion];
4.      else {
5.          aux = mayor (numeros, posicion - 1);
6.          if (numeros [posicion] > aux) return numeros [posicion];
7.          else return mayor (numeros, posicion - 1);
8.      }
9.      }
```

Citar

Recursividad

1.1. Introducción.

El concepto de recursividad va ligado al de repetición. Son recursivos aquellos algoritmos que, estando encapsulados dentro de una función, son llamados desde ella misma una y otra vez, en contraposición a los algoritmos iterativos, que hacen uso de bucles while, do-while, for, etc.

1.2. Definición.

Algo es recursivo si se define en términos de sí mismo (cuando para definirse hace mención a sí mismo). Para que una definición recursiva sea válida, la referencia a sí misma debe ser relativamente más sencilla que el caso considerado.

1.3. Elementos de la Recursión

1.3. 1. Axioma

Es un caso donde el problema puede resolverse sin tener que hacer uso de una nueva llamada a sí mismo. Evita la continuación indefinida de las partes recursivas.

1.3.2. Formula recursiva

Relaciona el resultado del algoritmo con resultados de casos más simples. Se hacen nuevas llamadas a la función, pero están más próximas al caso base.

Por ejemplo: **El factorial de un número**

```
factorial(0) -> 1
factorial(1) -> 1*factorial(0)
factorial(2) -> 2*factorial(1)
factorial(3) -> 3*factorial(2)
...
factorial(N) -> 3*factorial(N-1)
```

En la resolución de algoritmos recursivos es imprescindible encontrar estos dos elementos.

1.4. Tipos de recursión

1.4.1. Recursividad simple

Aquella en cuya definición sólo aparece una llamada recursiva. Se puede transformar con facilidad en algoritmos iterativos.

1.4.2. Recursividad múltiple

Se da cuando hay más de una llamada a sí misma dentro del cuerpo de la función, resultando más difícil de hacer de forma iterativa. Un ejemplo típico es la función de fibonacci

1.4.3. Recursividad anidada

En algunos de los argumentos de la llamada recursiva hay una nueva llamada a sí misma. La función de Ackermann se define por recursividad como sigue:

1.4.4. Recursividad cruzada o indirecta

Son algoritmos donde una función provoca una llamada a sí misma de forma indirecta, a través de otras funciones.

OHK

« Última modificación: 15 Marzo 2011,
23:52 por ohk »

  En línea

Y es que a veces pienso que si no estuviera loco no podría salir adelante.
Lo que no se es capaz de dar, en realidad no se posee, uno es poseído por ello.

@ohk



Re: Ejercicios Revursivos en Java y sus Soluciones
« Respuesta #1 en: 11 Octubre 2008, 03:47 »



Desconectado

Alguien se anima a resolver el **ejercicio 14** ?

Mensajes:
1.728

OHK



Yo lo que
quiero que me
salga bien es
la vida.

« Última modificación: 12 Octubre 2008,
05:53 por ohk »

  En línea



Y es que a veces pienso que si no estuviera loco no podría salir adelante.
Lo que no se es capaz de dar, en realidad no se posee, uno es poseído por ello.

Amerikano|CIs



Re: Ejercicios Recursivos en Java y sus Soluciones
« Respuesta #2 en: 13 Noviembre 2008, 02:17 »





Desconectado

lo intentare hacer pero necesito tiempo, por que ando ocupado.

Mensajes:
789

amerikano

  En línea

[Beyond This
Life]



アメリカノ



Mi blog:

<http://amerikanocls.blogspot.com>

Amerikano|Cls



Re: Ejercicios Recursivos en Java y sus Soluciones

« **Respuesta #3** en: 27 Noviembre 2008, 22:30 »



Desconectado

Mensajes:
789

[Beyond This
Life]

Oye una recomendación, el de fibonnaci no es conveniente hacerlo por recursividad, por lo siguiente, y es que las llamadas recursivas se dividen en dos lo cual conlleva a repetir un mismo proceso mas de dos veces y eso le resta velocidad y eficiencia era solo eso jeje.

salu2

En línea



アメリカノ



Mi blog:

<http://amerikanocls.blogspot.com>

Amerikano|Cls



Re: Ejercicios Recursivos en Java y sus Soluciones

« **Respuesta #4** en: 4 Diciembre 2008, 18:38 »



Desconectado

Mensajes:
789

[Beyond This
Life]

[Cita de: ohk en 11 Octubre 2008, 03:19](#)

Planteamiento:

Ejercicio 14. Programar un algoritmo recursivo que permita resolver la siguiente matriz:

```
1 1 1 1 1
1 2 2 2 2
1 2 4 4 4
1 2 4 8 8
1 2 4 8 16
```

Solución:

Código

1. No solucionado...

Bueno yo ya solucione el numero 14 jeje estaba facilito solo que no tenia tiempo de hacerlo, aca va:

Código

```
1. public class MatrizRecursividad {
2.
3.     private static int a=0, aux=1, b=0; //Declaramos los datos
4.     private static int[][] matriz = new int[6][6]; //La matriz
5.
6.     public static void main(String[] args) {
7.
8.         llenarMatriz(matriz, a, b); //Iniciamos el llamado
9.         imprimir(); //imprimimos la matriz
10.
11.     }
12.
13.     public static void llenarMatriz(int matriz[][], int i, int j) {
14.
15.         if(j > matriz.length - 1){ //Si llego a la ultima
```

```

16.         i++;
17.         j=0;
18.         aux++;
19.     }
20.     if(i < matriz.length){ // compara que no hallamos
21.
22.         if(i==(aux-1) && j >= (aux-1)){ //comprue
23.             if(i==0)// si es la primera fila
24.                 matriz[i][j] = matriz[i][
25.             else
26.                 matriz[i][j] = matriz[i][
27.                 llenarMatriz(matriz, i , j+1);
28.         }
29.         else{ //si no, asignamos los valores ante
30.             if(j==0)// comprobamos si es el p
31.                 matriz[i][j] = j+1;
32.             else
33.                 matriz[i][j] = matriz[i-1
34.                 llenarMatriz(matriz, i, j+1);
35.         }
36.     }
37. }
38.
39. public static void imprimir(){ //este metodo nos imprime
40.
41.     for(int i=0; i< matriz.length; i++){
42.         for(int j=0; j< matriz.length; j++){
43.             System.out.print(matriz[i][j] + "
44.         }
45.         System.out.print("\n");
46.     }
47. }
48.
49. }
50.

```

Bueno cualquier duda comenten que pa eso estoy 🤖.

salu2

amerikano

 En línea

アメリカノ



Mi blog:

<http://amerikanocls.blogspot.com>

@ohk



Re: Ejercicios Recursivos en Java y sus Soluciones

« **Respuesta #5 en:** 4 Diciembre 2008, 20:19 »



Desconectado

Mensajes:
1.728

@AmeRiK@nO, tu ejercicio corre perfectamente, y el código no es difícil de interpretar, porque haces código limpio.

Ahora mismo lo agrego tu ejercicio a el espacio que falta.
Gracias por el aporte
Un saludo

 En línea



Y es que a veces pienso que si no estuviera loco no podría salir adelante.
Lo que no se es capaz de dar, en realidad no se posee, uno es poseído por ello.

Yo lo que
quiero que me
salga bien es
la vida.



Amerikano|Clis



Re: Ejercicios Recursivos en Java y sus Soluciones

« **Respuesta #6** en: 5 Diciembre 2008, 01:49 »



Desconectado

Mensajes:
789

[Beyond This
Life]

jeje debes corregir esto:

Código

```
1. if(i==0)// si es la primera fila ingresamos aux=1
2. matriz[i][j] = matriz[i][j]=aux;
```

Vez, lo puse dos veces sin querer, ahora pon solo uno:

Código

```
1. if(i==0)// si es la primera fila ingresamos aux=1
2. matriz[i][j] =aux;
```

Se me coló uno sin darme cuenta , pero así y todo funcionó , solo era eso.
salu2

En línea

アメリカノ



Mi blog:

<http://amerikanocls.blogspot.com>

@ohk



Re: Ejercicios Recursivos en Java y sus Soluciones

« **Respuesta #7** en: 5 Diciembre 2008, 03:18 »



Desconectado

Mensajes:
1.728



Yo lo que
quiero que me
salga bien es
la vida.



[Cita de: AmeRik@nO en 5 Diciembre 2008, 01:49](#)

jeje debes corregir esto:

Código

```
1. if(i==0)// si es la primera fila ingresamos aux=1
2. matriz[i][j] = matriz[i][j]=aux;
```

Vez, lo puse dos veces sin querer, ahora pon solo uno:

Código

```
1. if(i==0)// si es la primera fila ingresamos aux=1
2. matriz[i][j] =aux;
```

Se me coló uno sin darme cuenta , pero así y todo funcionó , solo era eso.

salu2



Da igual, además quien lo use se dará cuenta


Y es que a veces pienso que si no estuviera loco no podría salir adelante.
Lo que no se es capaz de dar, en realidad no se posee, uno es poseído por ello.

juanch77

Re: Ejercicios Recursivos en Java y sus Soluciones
« Respuesta #8 en: 7 Diciembre 2008, 01:35 »



Desconectado

De todos modos no tiene error, ni sintactico ni semantico. Mensajes:
455rie con
demencia**think nonlinearly****pixzeto**

Desconectado



Re: Ejercicios Recursivos en Java y sus Soluciones
« Respuesta #9 en: 11 Junio 2009, 02:03 »

Mensajes: 1

Estuve practicando con estos ejercicios y parece que el método para invertir un número no funciona bien.

Por ejemplo, si le entregamos el 32, daría:
 $2 + \text{invertir}(3) \cdot 10 = 2 + 3 \cdot 10 = 32$





Código:

```
int invertir (int n)
{
    if (n < 10)          //caso base
        return n;
    else
        return (n % 10) + invertir (n / 10) * 10;
}
```

Pero lo hice así y funciona:

Código:

```
public int invertirNumero(int numero){
    if(numero<10){
        return numero;
    }else{
        int contador = 0;
        int aux = numero;
        while(aux/10!=0){
            contador++;
            aux = aux/10;
        }
        return (int) (Math.pow(10, contador))*(numero%10) +
this.invertirNumero(numero/10);
    }
}
```

Páginas: [1] 2 3 4 5  ir arriba responder imprimirIr a: **Mensajes similares**

Asunto

Iniciado por Respuestas Vistas Último mensaje

		Soluciones ejercicios C « 1 2 » Ejercicios	Cobac	13	42,782	7 Agosto 2005, 21:55 por Cobac	
		soplo pon las soluciones a los ejercicios de VB pliss Ejercicios	Codename!!	4	4,242	18 Julio 2004, 08:31 por Codename!!	
		Ejercicios Java Ejercicios	Lorena88	1	4,987	21 Enero 2009, 01:42 por Amerikano Cls	
		ejercicios + soluciones en java « 1 2 3 » Java	.rn3w.	25	14,999	14 Septiembre 2011, 19:01 por .rn3w.	
		Regalo para los que empezar a programar (triangulos recursivos): Java	4nc3str4l	2	1,851	25 Enero 2014, 10:01 por DarkSorcerer	

DriverLandia	MundoDivx	Hispabyte	Truzone
Yashira.org	indetectables.net	Seguridad Colombia	Seguridad Informática
Internet móvil	ADSL	eNYe Sec	Seguridad Wireless
Underground México	El Lado del Mal	Blog Uxio	thehackerway
Tienda Wifi	underc0de		

Todas las webs afiliadas están libres de publicidad engañosa.

Aviso Legal - Powered by SMF 1.1.21 | SMF © 2006-2008, Simple Machines