C ----------------------------------------------------------------------------------------

Pseudocodigo ordenar numeros

bool ordenado = false;

while(!ordenado) {

ordenado = true;

for ( int i = 0; i < arraySize - 1; i++) {

if (array[i] > array[i+1]) {

intercambiar(array[i], array[i+1])

ordenado = false;

}

}

}

*1ºIteracion:* 4, 1, 3, 2 *2ºIteracion:* 1, 4, 3, 2 *3ºIteracion:* 1, 3, 4, 2 *4ºIteracion:* 1, 3, 2, 4 *FIN DE PRIMERA ITERACION (WHILE)* *1ºIteracion:* 1, 3, 2, 4 *2ºIteracion:* 1, 2, 3, 4 *FIN DE SEGUNDA ITERACION (WHILE)* *1ºIteracion:* 1, 2, 3, 4 *TERCERA ITERACION (WHILE):* Recorre el for, pero no se mete nunca en el if, asi que ordenado se queda true

Convertir un unico char en int: int comoEntero = caracter - '0';

Recorrer array en c:

for (int i = 0; i < sizeof(miArray) / sizeof(miArray[0]); i++)

{

printf("%d ", miArray[i]);

}

Coger valor que dejo el hijo en status:

Hijo: exit(resultado);

Pdre:

int statusResultado;

// Esperar al proceso hijo y coger su estado(resultado de la operacion) desde direccion de memoria

wait(&statusResultado);

// Mostrar el resultado

printf("Resultado: %d\n", WEXITSTATUS(statusResultado));

//IMportante porque sino te devuelve un numero no interpretable, y no el valor del exit

FUNCION ESPRIMO()

bool esPrimo(int numero)

{

// Si el número es menor o igual a 1, no es primo

if (numero <= 1)

{

return false;

}

// Verificar si el número es divisible por algún otro número diferente de 1 y él mismo

for (int i = 2; i <= numero/2 ; i++)

{

if (numero % i == 0)

{

return false; // El número no es primo

}

}

return true; // El número es primo

}

LEER DE PIPE EN C

while (read(pipe\_fd1[READ], &numeroPar, sizeof(numeroPar)) > 0)

{ // mientras pueda seguir leyendo

printf("Soy el hijo 1, he recibido: %d\n", numeroPar);

}

PISTA:

* Usar dos tuberias Ej7 entrenamientoC

JAVA -----------------------------------------------------------------------------------------------

HACER HILO

* Hacer hilo extendiendo de Thread (clase de la API de Java)

public class MiHilo extends Thread{

public void run() {

System.out.println("Hilo ejecutándose");

}

}

public class Main {

public static void main(String[] args) {

MyThread t = new MyThread();

t.start();

}

}

* Hacer hilo implementando Runnable (interfaz de la API de Java)

class MyRunnable implements Runnable {

public void run() {

System.out.println("Hilo ejecutándose");

}

}

public class Main {

public static void main(String[] args) {

Thread t = new Thread(new MyRunnable());

t.start();

}

}

CREAR HILO CON LAMBDA

public class CrearHiloLambda {

public static void main(String[] args) {

Runnable runnable = () -> {

System.out.println("Hola mundo");

};

Thread t = new Thread(runnable);

t.start();

}

}

PISTA:

* Join(); en hilos de Java es homologo de wait();