Ejercicios arrays 3 (opcionales)

1.- Partiendo de dos arrays de cualquier longitud y que pueden ser de longitud diferente crear un tercer array con las posiciones intercaladas de los dos primeros, cuando se hayan acabado las posiciones de un array se pondrán a continuación las posiciones del otro. Por ejemplo {1, 2, 3} y {4, 5, 6, 7, 8} daría como resultado {1, 4, 2 , 5, 3, 6, 7, 8}.

Probarlo con {4, 7, 4, 7, 3, 23, -4, 12, 6} y {5, 7, 8, 9, 10}

**int** a[]= {4, 7, 4, 7, 3, 23, -4, 12, 6};

**int** b[]= {5, 7, 8, 9, 10};

**int** counta=0;

**int** countb=0;

**int** c[]=**new** **int** [a.length+b.length];

**int** countc=0;

**while** (counta<a.length && countb<b.length) {//para intercalar las posiciones

c[countc++]=a[counta++];

c[countc++]=b[countb++];

}

**while**(countb<b.length) {

c[countc++]=b[countb++];

}

**while**(counta<a.length) {

c[countc++]=a[counta++];

}

System.***out***.println(Arrays.*toString*(c));

2.- Sumar 3 arrays de longitud diferentes. El array solución será del tamaño del array más grande, y en las posiciones que no existan de los arrays más cortos se considererá que hay un 0.

Probarlo con {4, 7, 9, -6, 5}, {5, 7, 1, 7, 2, 7, 23}, {5, 6, 10}

**int** a[]= {4, 7, 9,-6, 5};

**int** b[]= {5, 7, 1, 7, 2, 7, 23};

**int** c[]= {5, 6, 10};

**int** sol[]=**new** **int**[b.length];

**for** (**int** i=0;i<b.length;i++) {

**if**(i<c.length) {

sol[i]=a[i]+b[i]+c[i];

}

**if**(i<a.length && i>=c.length) {

sol[i]=a[i]+b[i];

}

**if** (i>=a.length) {

sol[i]=b[i];

}

}

System.***out***.println(Arrays.*toString*(sol));

3.- Dado un array con elementos entre -3 y 3 (cero incluido) obtener un array solución de la siguiente manera.

- Un -3 se transforma en tres -3

- Un -2 se transforma en dos -2

- Un -1 se transforma en un -1

- Un 1 se transforma en un 1

- Un 2 se transforma en dos 2

- Un 3 se transforma en tres 3

Cualquier valor diferente de los anteriores desaparece.

Ejemplo, el array {-2,0, 3, 5, 2} daría : {-2,-2, 3, 3, 3, 2, 2}.

Probarlo con {2, -3, 0, 1, 2, -2}

**int** a[]= {2,-3,0,1,2,-2};

**int** count=0;

**for** (**int** i=0;i<a.length;i++) {

**if**(a[i]==3 || a[i]==-3) {

count+=3;

}

**if**(a[i]==2 || a[i]==-2) {

count+=2;

}

**if**(a[i]==1 || a[1]==-1) {

count++;

}

}

**int** nuevo[]=**new** **int**[count];//nuevo array

**int** pos=0;//posicion en el nuevo array

**for** (**int** i=0;i<a.length;i++) {

**if** (a[i]==3) {

nuevo[pos]=3;

nuevo[pos+1]=3;

nuevo[pos+2]=3;

pos+=3;

}

**else** **if** (a[i]==2) {

nuevo[pos]=2;

nuevo[pos+1]=2;

pos+=2;

}

**else** **if** (a[i]==1) {

nuevo[pos]=1;

pos++;

}

**else** **if** (a[i]==-1) {

nuevo[pos]=-1;

pos++;

}

**else** **if** (a[i]==-2) {

nuevo[pos]=-2;

nuevo[pos+1]=-2;

pos+=2;

}

**if** (a[i]==-3 ) {

nuevo[pos]=-3;

nuevo[pos+1]=-3;

nuevo[pos+2]=-3;

pos+=3;

}

}

System.***out***.println(Arrays.*toString*(nuevo));

 4.- declarar un int a= valor inicial, int b=valor final;

Para cualquier array obtener el subarray que se encuentra entre la posición del valor a más a la izquierda y el valor de la posición b más a la derecha.

Ejemplo, con a= 4, y b = 1 y el array { 3, 6, 4, 6, 8, 1, 6, 9, 10} la solución será {4, 6, 8,1}

Si no se da la condiciones de que haya a o b en el array, o b está más a la izquierda que a deberá de indicarlo mediante un mensaje y no resolver el array solución.

**int** a=2;

**int** b=4;

**int** c[]={ 3, 6, 4, 6, 8, 1, 6, 9, 10} ;

**int** posa=a-1;

**int** posp=b+1;

**int** d[]= **new** **int**[posp-posa+1];

**int** count=0;

**for** (**int** i=posa; i<(posp+1);i++) {

d[count]=c[i];

count++;

}

System.***out***.println(Arrays.*toString*(d));

}

}