

Lógicamente, hay muchas soluciones válidas, aquí hay una solución de tantas.

EJERCICIO1.

Una unidad astronómica, abreviadamente se escribe como *ua* y es la distancia media entre la Tierra y el Sol, equivalente a 149 597 870 km. Se usa para expresar de forma cómoda las distancias entre el sol y los planetas del sistema solar. La distancia expresada en *ua* del sol a la tierra se toma como referencia y es de 1.00ua. Del sol a Plutón es 39.44ua. Calcula cuántos km separan a plutón del sol. Para observar que double es más preciso que float haz el cálculo con las dos aritméticas de forma que obtengamos la siguiente salida.

run:

km del sol a plutón utilizando float: 5.90014E9

km del sol a plutón utilizando double: 5.900139992799999E9

SOLUCIÓN

```
class Unidad1 {
    public static void main(String[] args) {
        final int UA_EN_KM= 149597870;
        float plutonFloat=UA_EN_KM*39.44f;
        double plutonDouble=UA_EN_KM*39.44;
        System.out.println("km del sol a plutón utilizando float: " +plutonFloat);
        System.out.println("km del sol a plutón utilizando double: " +plutonDouble);
    }
}
```

EJERCICIO2.

Queremos calcular cuántos kilómetros tiene un año luz. Teniendo en cuenta los siguientes datos:

- un año luz son los kilómetros que recorre la luz en un año.
- la velocidad de la luz, redondeando, es 300000km/s
- Una año, redondeando, tiene 365 días.

Escribimos el siguiente código

```
class Unidad1 {
    public static void main(String[] args) {
        final int VELOCIDAD_LUZ=300000;
        final int SEGUNDOS_UN_AÑO=365*24*60*60;
        int añoLuz=SEGUNDOS_UN_AÑO*VELOCIDAD_LUZ;
        System.out.println("UN AÑO LUZ TIENE " +añoLuz+"km.");
    }
}
```

y obtenemos la siguiente salida:

run:

UN AÑO LUZ TIENE -1012953088km.

Se pide:

- explica porque el código anterior produce una salida incorrecta.
- Reforma el código anterior para que obtener el resultado correcto

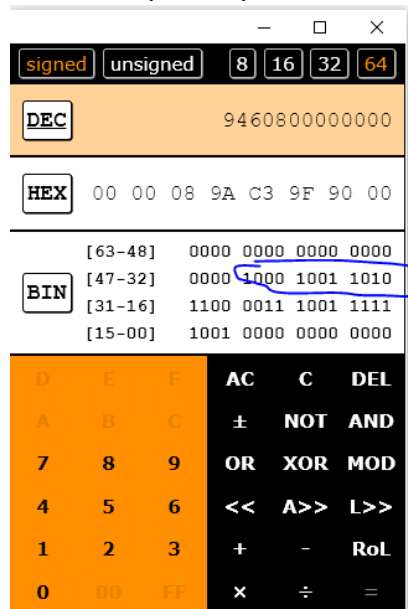
Explicación con calculadora

$365 \times 24 \times 60 \times 60 \times 300000 = 9_460_800_000_000$ (se usa guión bajo por legibilidad). El mayor valor para el tipo int es **2_147_483_647** por tanto 9 460 800 000 000 excede el valor máximo representable

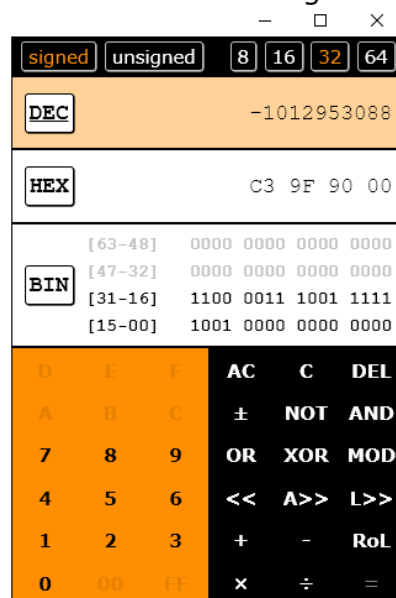
9_460_800_000_000
2_147_483_647

, por tanto hay truncamiento de bits y el valor que obtenemos es erróneo. Para entenderlo nos podemos ayudar de la calculadora

Observa que hay 1s más allá de los 32 bits



Si paso de 64 bits a 32 bits, ahora el bit 31 (el 2^{30}) es un 1 y por tanto se interpreta como un número negativo en complemento a 2



En java la expresión `SEGUNDOS_UN_AÑO*VELOCIDAD_LUZ`; es de tipo `int` y por tanto se almacena en 32 bits y se produce el truncamiento

Observa que si declaro la variable `añoLuz` como `long`

```
class Unidad1 {
    public static void main(String[] args) {
        final int VELOCIDAD_LUZ=300000;
        final int SEGUNDOS_UN_AÑO=365*24*60*60;
        long añoLuz=SEGUNDOS_UN_AÑO*VELOCIDAD_LUZ;
        System.out.println("UN AÑO LUZ TIENE " +añoLuz+"km.");
    }
}
```

El problema persiste ya que el truncamiento de bits ya se produjo al evaluar la expresión `SEGUNDOS_UN_AÑO*VELOCIDAD_LUZ`. recuerda que las expresiones tienen un tipo y en este caso le corresponde tipo `int`!. Por lo tanto, tenemos que conseguir que esta expresión sea `long`. Se puede hacer de muchas formas, por ejemplo, basta con que un operando sea `long` y así toda la expresión es `long`.

```
public class Unidad1 {
    public static void main(String[] args) {
        final int VELOCIDAD_LUZ=300000;
        final int SEGUNDOS_UN_AÑO=365*24*60*60;
        long añoLuz=(long)SEGUNDOS_UN_AÑO*VELOCIDAD_LUZ;
        System.out.println("UN AÑO LUZ TIENE " +añoLuz+"km.");
    }
}
```

run:

UN AÑO LUZ TIENE 94608000000000km.

EJERCICIO3.

Averigua imprimiendo por pantalla

- que carácter pertenece a los siguientes códigos unicode (en decimal): 97 y 225.
- Cuáles son los códigos Unicode de los siguientes caracteres: e,é, ñ, ?, y ¿.

Lógicamente, el ejercicio debes realizarlo sin consultar una tabla unicode . Utiliza la relación `char/int` de java.

la salida debe ser parecida a lo siguiente:

caracter unicode en decimal 97: a
caracter unicode en decimal 225: á
caracter unicode e en decimal: 101
caracter unicode é en decimal: 233

caracter unicode ñ en decimal: 241
caracter unicode ? en decimal: 63
caracter unicode ¿ en decimal: 191

SOLUCIÓN:

```
public class Unidad1 {  
    public static void main(String[] args) {  
        System.out.println("caracter unicode en decimal 97: " + (char)97);  
        System.out.println("caracter unicode en decimal 225: " + (char)225);  
        System.out.println("caracter unicode e en decimal: " + (int)'e');  
        System.out.println("caracter unicode é en decimal: " + (int)'é');  
        System.out.println("caracter unicode ñ en decimal: " + (int)'ñ');  
        System.out.println("caracter unicode ? en decimal: " + (int)'?');  
        System.out.println("caracter unicode ¿ en decimal: " + (int)'¿');  
    }  
}
```

EJERCICIO4:

Tengo una variable x. La inicializas al valor que tú quieras. El resto del código comprueba si x es par en cuyo caso la incrementa en 1 e imprime su nuevo valor , y en caso contrario imprime "sin cambios". Es obligatorio escribir el programa con el operador condicional.

```
class Unidad1 {  
    public static void main(String[] args) {  
        int x=4;  
        System.out.println(x%2==0?++x:"Sin cambios");  
    }  
}
```

EJERCICIO5.

El siguiente programa presenta errores de compilación. Corrígelo utilizando el operador de cast

```
class Unidad1{  
    public static void main(String args[]){  
        int i;  
        float f;  
        char c;  
        i=65.3;  
        f=65.3;  
        c=65.3;  
        System.out.println(i+" "+f+" "+c+" "+(int)c);  
    }  
}
```

SOLUCIÓN:

```
class Unidad1{  
    public static void main(String args[]){  
        int i;  
        float f;  
        char c;  
        i=(int)65.3;  
        f=(float)65.3;  
        c=(char)65.3;  
        System.out.println(i+" "+f+" "+c+" "+(int)c);  
    }  
}
```

EJERCICIO6.

Utilizando el operador de desplazamiento correspondiente, divide 32 por 2 e imprime el resultado por pantalla.

SOLUCIÓN:

```
class Unidad1{
    public static void main(String args[]){
        int desplazamiento=32>>1;
        System.out.println("32 DIVIDIDO POR 2 ES "+ desplazamiento);
    }
}
```

O BIEN

```
class Unidad1{
    public static void main(String args[]){
        System.out.println("32 DIVIDIDO POR 2 ES "+ (32>>1));
    }
}
```

Debes de entender la necesidad de escribir en este último ejemplo la expresión 32>>1 entre paréntesis. Comprueba que pasa si los quitas. Comprueba también que el + tiene mayor prioridad que el >> en una tabla de precedencia de operadores java

EJERCICIO7.

Utilizando el operador de decremento -- , obtén código equivalente a:

```
class Unidad1{
    public static void main(String args[]){
        int x=4;
        int y;
        x=x-1;
        y=x*2;
        System.out.println("x: "+ x+ " y:"+y);
    }
}
```

SOLUCIÓN:

Varias posibilidades, algunas...

```
class Unidad1{
    public static void main(String args[]){
        int x=4;
        int y;
        x--;
        y=x*2;
        System.out.println("x: "+ x+ " y:"+y);
    }
}
```

OTRA

```
class Unidad1{
    public static void main(String args[]){
        int x=4;
        int y;
        y=-x*2;
        System.out.println("x: "+ x+ " y:"+y);
    }
}
```

OTRA MÁS LIOSA PERO POSIBLE Y EQUIVALENTE

```
class Unidad1{
    public static void main(String args[]){
        int x=4;
        System.out.println("x: "+ --x + " y:" +x*2);
    }
}
```

OTRA MAL

```
class Unidad1{
    public static void main(String args[]){
        int x=4;
        System.out.println("x: "+ x-- + " y:" +x*2);
    }
}
```

EJERCICIO8.

Utilizando los operadores de asignación combinados += -= /= *= escribe un código equivalente al siguiente:

```
class Unidad1{
    public static void main(String args[]){
        int a=10;
        System.out.println(a);
        a=a/2;
        System.out.println(a);
        a=a+3;
        System.out.println(a);
        a=a*2;
        System.out.println(a);
        a=a-2;
        System.out.println(a);
    }
}
```

SOLUCION:

```
class Unidad1{
    public static void main(String args[]){
        int a=10;
        System.out.println(a);
        a/=2;
        System.out.println(a);
        a+=3;
        System.out.println(a);
        a*=2;
        System.out.println(a);
        a-=2;
        System.out.println(a);
    }
}
```

EJERCICIO 9.

Escribe un programa que obtenga como resultado la siguiente ejecución:

```
las comillas dobles son así "
```

SOLUCIÓN:

```
class Unidad1{
```

```

public static void main(String args[]){
    System.out.println("Las comillas dobles son así \"");
}
}

```

EJERCICIO 10.

```

class Unidad1{
    public static void main(String args[]){
        System.out.println("falso \t||\tfalso \t=\t" +(false||false) );
        System.out.println("falso \t||\tverdadero\t=\t" +(false||true) );
        System.out.println("verdadero\t||\tfalso \t=\t" +(true||false) );
        System.out.println("verdadero\t||\tverdadero\t=\t" +(true||true) );
    }
}

```

EJERCICIO 11.

```

class Unidad1{
    public static void main(String args[]){
        System.out.println((1 << 0)+" en binario\t"+ Integer.toBinaryString(1 << 0));
        System.out.println((1 << 1)+" en binario\t"+ Integer.toBinaryString(1 << 1));
        System.out.println((1 << 2)+" en binario\t"+ Integer.toBinaryString(1 << 2));
        System.out.println((1 << 3)+" en binario\t"+ Integer.toBinaryString(1 << 3));
        System.out.println((1 << 4)+" en binario\t"+ Integer.toBinaryString(1 << 4));
        System.out.println((1 << 5)+" en binario\t"+ Integer.toBinaryString(1 << 5));
        System.out.println((1 << 6)+" en binario\t"+ Integer.toBinaryString(1 << 6));
        System.out.println("etc..");
    }
}

```

EJERCICIO 12.

De Alejandro Oubiña 2016/2017

```

class Rumore{
    public static void main(String args[]){
        int amigos=2;
        amigos=amigos<<1;
        System.out.println("El primer día lo saben "+amigos+" \"amigos\".");
        amigos=amigos<<1;
        System.out.println("El segundo día lo saben "+amigos+" \"amigos\".");
        amigos=amigos<<1;
        System.out.println("El tercer día lo saben "+amigos+" \"amigos\".");
        amigos=amigos<<1;
        System.out.println("El cuarto día lo saben "+amigos+" \"amigos\".");
        amigos=amigos<<1;
        System.out.println("El quinto día lo saben "+amigos+" \"amigos\".");
        amigos=amigos<<1;
        System.out.println("El sexto día lo saben "+amigos+" \"amigos\".");
        amigos=amigos<<1;
        System.out.println("El séptimo día descans... digooo lo saben "+amigos+" \"amigos\".");
        System.out.println("Mejor lo dejamos aquí, ya lo sabe todo el mundo.");
    }
}

```

EJERCICIO 13.

De un alumno del 16/17 no recuerdo su nombre

```

class SeisGrados{
    public static void main(String[] args){
        long ungrado=100; long dosgrado=0; long tresgrado=0; long cuatrogrado=0; long cincogrado=0; long seisgrado=0;
    }
}

```

```

//le asignamos long a todas las variables porque al multiplicar "ungrado" por "cincogrado" o "seisgrado"
//el resultado debe ser guardado en un long.
//Podríamos dejar las variables "dosgrado", "tresgrado" y "cuatrogrado" como int, pero he preferido dejarlas todas del mismo tipo.
    dosgrado=ungrado*ungrado;
    tresgrado=dosgrado*ungrado;
    cuatrogrado=tresgrado*ungrado;
    cincogrado=cuatrogrado*ungrado;
    seisgrado=cincogrado*ungrado;

    System.out.println("Yo conozco a "+ungrado+" personas");
    System.out.println("Esas "+ungrado+" conocen a otras "+ungrado+" personas cada una, así que en segundo grado tengo acceso a "+dosgrado+"
personas");
    System.out.println("Esas "+dosgrado+" conocen a otras "+ungrado+" personas cada una, así que en tercer grado tengo acceso a "+tresgrado+"
personas");
    System.out.println("Esas "+tresgrado+" conocen a otras "+ungrado+" personas cada una, así que en cuarto grado tengo acceso a "+cuatrogrado+"
personas");
    System.out.println("Esas "+cuatrogrado+" conocen a otras "+ungrado+" personas cada una, así que en quinto grado tengo acceso a "+cincogrado+"
personas");
    System.out.println("Esas "+cincogrado+" conocen a otras "+ungrado+" personas cada una, así que en sexto grado tengo acceso a "+seisgrado+"
personas");
    System.out.println("¡Enhorabuena! ¡Acabas de preguntarle la hora al presidente de los Estados Unidos!");
}
}

```

Nota de profesor: lee el comentario que dejó el alumno, y observa que si cuatrogrado fuera int la expresión cuatrogrado*ungrado habría que pasarla de alguna manera a long o ya obtenemos el resultado erróneo

Ejercicio 14:

GONZALO GOMEZ RIAL alumno del 16/17

```

public class Unidad1 {

    public static void main(String[] args) {

        int pesoElefante = 5300;
        int nElefantes = 15;
        int pesoTotal = nElefantes*pesoElefante;

        float grosorHilo = (pesoTotal/2400f);
        System.out.println("Necesitaremos un hilo de "+grosorHilo+" cm para soportar el peso de los elefantes");

        float gHiloArana = 0.001f;
        int nAranas=15000;
        float totalAranas=(grosorHilo/gHiloArana)*nAranas;
        System.out.println("Necesitaremos ademas un total de "+totalAranas+" arañas");
    }

}

```

Nota del profesor: con float y double se pueden almacenar unas cantidades muy grandes pero siempre van a ser imprecisas. En este caso arañita arriba o abajo nos da igual ciertamente.

Ejercicio 15:

RUTH alumno del 16/17

Empleando operadores a nivel de bits, se pide que conviertas un número entero positivo a su correspondiente negativo.

```

class Unidad1{
    public static void main(String args[]){
        int iNumero = 45;
        System.out.println("El negativo de 45 es: "+((~iNumero)+1));
        /* Java trabaja con el Complemento a dos, por tanto para obtener un
        número negativo en complemento a dos primero se niegan todos los bits y luego
        se le suma 1 */
    }
}

```

Ejercicio 16:

JOAQUIN 16/17

```
class Unidad1 {  
    public static void main(String args[]){  
        byte numEnDecimal = 12;  
        String binarioOriginal = Integer.toBinaryString(numEnDecimal);  
        String a2= Integer.toBinaryString(numEnDecimal*-1);  
        String a1= Integer.toBinaryString((numEnDecimal*-1)-1);  
        System.out.println("El numero "+numEnDecimal+" se escribe en binario de esta manera: "+binarioOriginal);  
        System.out.println("El numero "+numEnDecimal+" se escribe en complemento a1 de esta manera: "+a1);  
        System.out.println("El numero "+numEnDecimal+" se escribe en complemento a2 de esta manera: "+a2);  
    }  
}
```

Ejercicio 17:

SAUL alumno del 16/17

NOTA DEL PROFESOR: Con el operador ^ se consigue la operacion XOR

Para entender algo mejor lo que pretende el alumno

https://es.wikipedia.org/wiki/Cifrado_XOR

```
class Unidad1 {  
  
    public static void main(String args[]) {  
        char cCaracter = 'x';  
        int iCodigoCifrado = 26;  
        System.out.println("El caracter " + cCaracter + " cifrado es:" + (char) ((int) cCaracter ^ iCodigoCifrado));  
    }  
}
```

Y estaría bien que además de cifrar, ilustra que el cifrado es útil descifrando con la misma clave y recuperando el caracter original

```
class Unidad1 {  
  
    public static void main(String args[]) {  
        char cCaracter = 'x';  
        int iCodigoCifrado = 26;  
        char caracterCifrado= (char) ((int) cCaracter ^ iCodigoCifrado);  
        System.out.println("El caracter " + cCaracter + " cifrado es:" + caracterCifrado);  
        char caracterDescifrado=(char) ((int)caracterCifrado^ iCodigoCifrado);  
        System.out.println("El caracter descifrado es " + caracterDescifrado);  
    }  
}
```

Ejercicio 18:

BRAIS Alumno del 16/17

Calculadora de subnet teniendo la ip y la máscara. Ejemplo 172.16.0.1/16

IP & Mask = SUBNET

Nota del profesor: este ejercicio está muy bien pero evidentemente sólo lo entenderán y aprovecharán aquellos alumnos que conocen el concepto de dirección IP, máscara y subred. La dirección IP la almacena en 4 variables int a las que aplica el & con la parte de máscara que le corresponde.

```
class Unidad1{  
    public static void main(String args[]){  
  
        int ip1 = 172; int ip2 = 16; int ip3 = 0; int ip4 = 1;  
        int mask1 = 255; int mask2 = 255; int mask3 = 0; int mask4 = 0;  
        int subnet1=ip1&mask1; int subnet2=ip2&mask2; int subnet3=ip3&mask3; int subnet4=ip4&mask4;  
  
        System.out.println("\n"+"SUBNET CALCULATOR: "+"\\n");  
        System.out.println("IP: "+ip1+"."+ip2+"."+ip3+"."+ip4);  
        System.out.println("MASK: "+mask1+"."+mask2+"."+mask3+"."+mask4);  
        System.out.println("SUBNET: "+subnet1+"."+subnet2+"."+subnet3+"."+subnet4+"\\n");  
    }  
}
```

```
}
```

Ejercicio 19: Miguel 2015/16

Nota del profesor: el calculo en pelines de población china es un añadido del profesor para forzar el uso del Long.

```
class Unidad1{
    public static void main(String args[]){
        byte p= 2;
        double x=2/p;
        System.out.println("En una cucharilla de azúcar hay: " + x + "pelines");
        int y=1000/p;
        System.out.println("En sal en el salero de mi casa 1000g hay: " + y + "pelines");
        int z=1000000/p;
        System.out.println("En camión arena (1 tonelada): " + z + "pelines");
        System.out.println("peso en kg de chinos en int(1 chino 60kg): "+ 60* 1_357_000_000);
        System.out.println("peso en kg de chinos en long(1 chino 60kg): "+ (long)60* 1_357_000_000);
        System.out.println("peso en gramos de chinos en long(1 chino 60kg): "+ (long)60* 1_357_000_000*1000);
        System.out.println("peso en pelines de chinos en long(1 chino 60kg): "+ (long)60* 1_357_000_000*1000/2);
    }
}
```

Los habitantes de china caben en un int, pero si suponemos que el peso de medio de un chino es 60kg el peso en kg de los chinos ya necesita un long.

1_357_000_000 habitantes de china
2_147_483_647

60* 1_357_000_000 inecesita long!

el peso en gramos ¿también cabe en long?

60* 1_357_000_000*1000=81420000000000 y esto cabe en long ya que

81420000000000 peso en gramos
9223372036854775807 valor max long

Ejercicio 20: Borja 2017/18

Haz con el operador condicional una forma de comparar a la gente que esté aprobada y la gente que este suspensa, y que además con el or les diga a la gente que tenga aprobado les mande un mensaje por su nota.

```
class T1_20{
    public static void main(String[] args){
        int n1=9;
        System.out.println(n1>=5?"Aprobado con un "+n1:"Suspenso con un "+n1);
        System.out.println(((n1==9)|(n1==10))?" Felicidades tienes un sobresaliente":" ");
        System.out.println(((n1==8)|(n1==7))?" Felicidades tienes un notable":" ");
        System.out.println((n1==6)" Felicidades tienes un bien esfuercete un poco más":" ");
        System.out.println((n1==5)" Felicidades tienes un suficiente esfuercete más":" ");
    }
}
```

Ejercicio 21: Lorena 2017/18

```

class EjercicioLorena{

    public static void main(String[] args) {
        int x = 2356;
        int posicion1 = 2;
        int posicion2 = 3;
        int posicion3 = 5;
        int posicion4 = 6;
        System.out.println(posicion1 + posicion3 == posicion2 + posicion4 ? "el numero x es divisible por 11" : "no es divisible por 11");
    }
}

```

Ejercicio 21:

Juan Ramón Bodelo

Nota del profesor: interesante porque es un ejemplo de que en una expresión con operador condicional se puede anidar otro operador condicional

```

class ejercicio20{
    public static void main(String[]args){
        int aforo=300;
        int guiris=301;
        String situacion= (guiris<aforo) ? "Se pueden meter más guiris"
        : (guiris==aforo) ? "Aforo completo!" : "OVERBOOKING!!! Hay que sacar guiris!";
        System.out.println(situacion);
    }
}

```

Ejercicio 23: Rubén López Mera

Nota del profesor: me hubiera gustado incluso sin operador condicional (ni IF), como simple ejemplo de expresión aritmética.

```

class Ejercicio{
    public static void main (String args[]){
        int mueble = 92;
        int tvP = 38;
        double tvCM;
        tvCM = tvP*2.54;
        System.out.println(tvCM>=mueble?"La television mide "+tvCM+"cm, por lo tanto NO cabe en su mueble de "+mueble+"cm":"La television mide "+tvCM+"cm, por lo tanto SI cabe en su mueble de "+mueble+"cm");
    }
}

```

Ejercicio 24: María Serafín

Nota del profesor: Un bonito ejemplo que usa cast y operador condicional. Adaptado para mostrar sólo el concepto de cast, es decir, simplificándolo, también estaría muy bien.

```

class Ejercicio20{
    public static void main(String args[]){
        /*EJERCICIO 20
        Te has hecho rico minando bitcoins y le debes dinero a tu amigo.
        Estás dispuesto a devolvérselo siempre y cuando lo que te pida no supere
        los 127€, que no te hiciste rico regalando dinero. Si no estás dispuesto

```

a pagárselo dale la vuelta a la situación intentado que sea él quien te deba dinero a ti. (Utilizando la misma variable, tampoco hay que ser rastrero que es tu amigo y a este paso te van a quedar pocos.)*/*

```
    short dineros=128;
    System.out.println("¿Puedes devolverme los "+ dineros+" ahora que te va bien?
Ando un poco justo este mes");
    System.out.println(dineros>127? "En realidad te debo "+ (byte)dineros+", así que
eres tu quien me debe dinero a mí":"¡Claro que sí amigo mío!");
    }
}
```