

Tema 2 Instrucciones de control y menús

Tabla de ejecución if

1. Hágase la tabla de ejecución para el siguiente programa cuyos datos de entrada son 7, 5 y 9. ¿Qué obtiene dicho programa para cualesquiera datos de entrada?

```
//Variables
int a = 0, b = 0, c = 0;
int mayor = 0;

//Programa
a = in.leerInt();
b = in.leerInt();
c = in.leerInt();

if(a>b) mayor = a;
else mayor = b;

if(mayor<c) mayor = c;

System.out.println(mayor);
```

2. Hágase la tabla de ejecución para el siguiente programa cuyos datos de entrada son 'K', 'F' y 'J'. ¿Qué obtiene dicho programa para cualesquiera datos de entrada?

```
//Variables
char a = '\0 ', b = '\0 ', c = '\0 ';

//Programa
a = in.leerChar("Escribe una letra en mayúscula:");
b = in.leerChar("Escribe una letra en mayúscula:");
c = in.leerChar("Escribe una letra en mayúscula:");

if(a<=b && b<=c) System.out.println(""+a+b+c);
else if(a<=c && c<=b) System.out.println(""+a+c+b);
else if(b<=a && a<=c) System.out.println(""+b+a+c);
else if(b<=c && c<=a) System.out.println(""+b+c+a);
else if(c<=a && a<=b) System.out.println(""+c+a+b);
else System.out.println(""+c+b+a);
```

Instrucciones condicionales

3. Hágase un programa que lea dos enteros y compruebe si el primero es o no mayor que el segundo. En la salida se mostrará “El primero es mayor que el segundo” o “El primero no es mayor que el segundo”. (MayorQue)

4. Hágase un programa que lea tres enteros positivos, y compruebe si alguno es suma de los otros dos. En la salida se mostrará “Ninguno es suma de los otros dos” o “ $8 = 2 + 6$ ” (en este último caso se introdujeron los números 2, 8 y 6) (SumaTres)
5. Léase un número entero. Se comprobará si dicho número es o no múltiplo de 20, y también se comprobará si está o no entre -100 y 100. En la salida se mostrará uno de los siguientes mensajes:
 1. “Es múltiplo de 20 y está entre -100 y 100”.
 2. “Es múltiplo de 20 y no está entre -100 y 100”.
 3. “No es múltiplo de 20 y está entre -100 y 100”-
 4. “No es múltiplo de 20 y no está entre -100 y 100”. (Condiciones)
6. Hágase un programa que compruebe si un año es bisiesto. En la salida se mostrará o “Es bisiesto” o “No es bisiesto”. (Bisiesto)

El algoritmo para obtener si un año es bisiesto es el siguiente: Inicialmente se almacena en una variable de salida que no es bisiesto; si es múltiplo de 4, se almacena en dicha variable que es bisiesto; si es múltiplo de 100 se almacena en dicha variable que no es bisiesto y si es múltiplo de 400, se almacena en dicha variable que es bisiesto. Por último, se muestra el valor de la variable de salida.

7. Se introduce una nota entre 0.0 y 10.0, y se traduce dicha nota de la siguiente forma:

NOTA	CALIFICACIÓN
Superior o igual a 8.5	Sobresaliente
Entre 7 y 8.4	Notable
Entre 6 y 6.9	Bien
Entre 5 y 5.9	Suficiente
Entre 3 y 4.9	Insuficiente
Entre 0 y 2.9	Muy deficiente

(TraducirNota)

8. Determinar si un número natural de 4 dígitos (entre 1000 y 9999) es capicúa. Se comprobará previamente que el número introducido tiene 4 dígitos. En la salida se mostrará uno de los siguientes mensajes:
 - a) “El número no tiene 4 dígitos”;
 - b) “Es capicúa”
 - c) “No es capicúa”. (Capicua)

9. Introducir tres enteros entre -100 y 100. De cumplir los tres números la condición anterior, se obtendrá la suma de sus inversos redondeado a 5 decimales; si no se mostrará el mensaje “Alguno de los números no está entre -100 y 100”. No se sumará el inverso de aquellos valores que sean 0. Por ejemplo, se introducen los enteros 20, -4 y 100, y se muestra el valor $1/20 + 1/-4 + 1/100 = -0.19000$. (Inversos)
10. Un número de 3 cifras es un número de Armstrong si la suma de sus dígitos elevado a 3 coincide con dicho número. Por ejemplo, $153 = 1+125+27$. Dado un número de 3 cifras determinar si es o no un número Armstrong. Se comprobará previamente que el número introducido tiene 3 dígitos. En la salida se mostrará uno de los siguientes mensajes:
 1. “El número no tiene 3 dígitos”
 2. “Es un número Armstrong”
 3. “No es un número Armstrong”. (Armstrong)
11. Dados tres números reales, se comprobará si alguno de ellos es semisuma de los otros dos. Se mostrará por salida que “Un número es semisuma de los otros dos” o “Ningún número es suma de los otros dos”. (Semisuma)
12. Leer tres enteros y mostrarlos ordenados. (Ordenados)
13. Se leen tres enteros y se comprobará si son o no consecutivos. Se mostrará en la salida “Son consecutivos” o “No son consecutivos”. (Consecutivos)
14. Se lee un número entero. Dicho número se incrementará en una unidad si es par, y se decrementará en una unidad si es impar. Se muestra por consola el nuevo valor del número. (IncrementarDecrementar)
15. Se leen tres enteros y un valor booleano para ordenarlos ascendentemente o descendentemente. Se mostrarán los valores leídos ordenados. (OrdenarAD)
16. Dado un número entero, se incrementará en 2 unidades si es par, en 3 unidades si es múltiplo de 3, en 5 unidades si es múltiplo de 5, y si no, se incrementará en una unidad. Sólo se realiza un incremento. (UnIncremento)
17. Dados dos números naturales se comprobará si el primer número es múltiplo del segundo, mostrando un texto del tipo “n es múltiplo de m” o “n no es múltiplo de m”, siendo n y m los números introducido. En el caso de que el segundo número sea 0, se indicará que no es posible que el segundo valor sea 0. (SerMultiple)
18. Permítase introducir cinco enteros y muéstrese el primer positivo introducido (0 si no hubiera ninguno) y el último negativo introducido (0 si no hubiera ninguno). (PrimerPositivoUltimoNegativo)

- 19.** Pérmítase introducir el peso de una persona en kg con 1 decimal (el valor siempre es superior a 1.0 e inferior a 140.0) y su altura en metros con dos decimales (el valor siempre es superior a 0,25 e inferior a 2.20). Si el peso estuviera fuera del rango permitido, se asignaría como peso el de 100, y se mostraría un mensaje indicándolo. Si la altura estuviera fuera del rango permitido, se asignaría como altura 1, y se mostraría un mensaje indicándolo. Se obtendrá el índice de masa corporal o IMC que se obtiene como el cociente entre el peso en kilogramos y el cuadrado de la altura en metros redondeado a 1 decimal, y clasifíquese según la siguiente tabla:

IMC	Clasificación
<18,5	Peso insuficiente
18,5-24,9	Normopeso
25-26,9	Sobrepeso grado I
27-29,9	Sobrepeso grado II (preobesidad)
30-34,9	Obesidad de tipo I
35-39,9	Obesidad de tipo II
40-49,9	Obesidad de tipo III (mórbida)
>50	Obesidad de tipo IV (extrema)

(Imc)

Instrucción switch

- 20.** Programa que lea una letra, y muestre si es la vocal ‘a’, ‘e’, ‘i’, ‘o’ o ‘u’ o que no es una vocal. (Vocal)
- 21.** Léase un número entre 1 y 7, ambos inclusive. Se traducirán al día de la semana (“Lunes”, “Martes”, “Miércoles”, “Jueves”, “Viernes”, “Sábado” y “Domingo”) los valores del 1 al 7, respectivamente. (DiaDeLaSemana)
- 22.** Resuélvase el ejercicio anterior traduciendo los números 1 al 12 a los meses del año. (Mes)
- 23.** Establézcase para cada mes del año (introducidos en forma de cadena: “Enero”, “Febrero”, ..., “Diciembre”) el número de días que tienen. No se tendrá en cuenta si el año es bisiesto. Se agruparán los meses con el mismo número de días. (NumeroDeDiasMes)
- 24.** Sea un número con entre 1 y 999, ambos inclusive. Se traducirá dicho número a números romanos. (DeDecimalARomano)

Reglas de los números romanos:

- Los símbolos básicos con su valor son I (1), V (5), X (10), L (50), C (100), D (500) y M (1000)
- Las letras de mayor a menor valor se van sumando. Ejemplo: VII vale 5+1+1
- No se pueden poner tres veces seguidas la misma letra.

- Las letras de menor a mayor valor se restan. Ejemplo: IV vale 5-1
- Sólo se pueden poner una letra I antes de las letras V y X; sólo una letra X antes de la L; y sólo una letra C antes de la D y la M.

Tabla de ejecución bucle while

25. Hágase la tabla de ejecución para el siguiente programa cuyos datos de entrada son 12 y 8. ¿Qué obtiene dicho programa para cualesquiera datos de entrada?

```
//Variables
int a = 0, b = 0, mcd = 0;

//Programa
a = in.leerInt("Entero: ",v->v>0);
b = in.leerInt("Entero: ",v->v>0);
mcm = b;

while(a % mcd != 0)
    mcd--;

System.out.println("Máximo común divisor: "+mcd);
```

Tabla de ejecución bucle do-while

26. Hágase la tabla de ejecución para el siguiente programa cuyo dato de entrada es 1901. ¿Qué obtiene dicho programa para cualquier dato de entrada?

```
//Variables
int n = 0, u = 0, con = 0;

//Programa
n = in.leerInt("Entero: ",v->v>0);

do
{
    u = n % 10;
    n = n / 10;
    if(u==0) con++;
}while(n!=0);

System.out.println("Número de ceros: "+con);
```

Bucles while y do-while

27. Leer números enteros hasta introducir un 0. El programa obtendrá la suma de todos los números positivos. (SumarPositivos)
28. Calcúlese el mínimo común múltiplo de dos números entre 1 y 100. Para obtener el mcm se tomará uno de los valores introducidos y se irá multiplicando por 1, 2, 3, etc, hasta que el valor obtenido sea múltiplo del otro número, que sería el valor buscado. (MinimoComunMultiplo)

29. Leer un número entero y obtener el número de dígitos que tiene. (ContarDigitos)
30. Introducir un número de segundos y generar un cronometro (minutos y segundos a partir de 00:00) hasta alcanzar el tiempo leído. (Cronometro)
31. Hágase un programa que permita al usuario adivinar un número aleatorio entre 0 y 100 en 10 intentos. (Adivinar)
32. Prográmese una carrera de 6 caballos. Ganará el primero que obtenga 20 avances. (CarreraDeCaballos)
33. Simúlese el juego del mastermind. Dicho juego consiste en adivinar un número de 4 dígitos mayor o igual que 1000. Hay un máximo de 10 intentos para adivinar el número. En cada intento se indica el número de muertos y heridos. Un muerto significa que se ha adivinado un dígito del número a adivinar (el de las unidades, decenas, centenas o unidades de mil), y un herido significa que se ha adivinado un dígito pero está en una posición incorrecta. Siempre primero se obtienen los muertos y luego los heridos. Por ejemplo, si el número adivinar fuera el 1891, y se propone el número 1919, entonces se diría que hay 1 muerto (el de las unidades de millar) y 2 heridos (el 1 de las decenas debería estar en las unidades y el 9 de las centenas o unidades debería estar en las decenas). (Mastermind)

Contadores

34. Introducir enteros hasta introducir un 0, y contabilizar cuántos de ellos son pares (el 0 no es par ni impar). (ContarPares)

Acumuladores

35. Introducir enteros hasta leer el 0 o un número positivo. El programa obtendrá el producto de los negativos. Si no se introduce ningún negativo, se muestra un mensaje indicándolo. (ProductoNegativos)

Señales

36. Permítase introducir enteros hasta introducir el 0. Obténgase la suma de los tres primeros pares introducidos, y el producto de los 3 últimos impares. Si no hubiera 3 pares o impares, como resultado se mostraría 0. (ParesImpares)

Bucle for

37. Se leerá el número de temperaturas a introducir (de ser 0 o negativo se establecerá a 5). Obténgase la media con dos decimales de las temperaturas introducidas por consola. (TemperaturasIntroducidas)

38. Considérese un número entre 0 y 10. Se mostrará la tabla de multiplicar de dicho número con el formato: $5 \times 3 = 15$ (desde 0 hasta 10 en líneas distintas). (TablaDeMultiplicar)

39. Se lee la cotización en bolsa de un determinado valor durante un número de días que se introduce por consola (de ser el número de días menor que 2 se establecerá a 5). Obténgase el último día que más subió y bajó con respecto al día anterior. (Bolsa)

NOTA Se utilizará el tipo “double” para declarar variables, pero para el cálculo de la diferencia entre la cotización actual y anterior, se utilizará la clase BigDecimal.

40. Calcular el factorial de un número entre 0 y 20. El factorial de un número se define del siguiente modo: $F(0) = 1$; $F(1) = 1$; $F(n) = n(n-1)(n-2) \dots 1$ siendo $n > 1$. (Factorial)

41. Dado dos enteros entre 0 y 5 (la base y el exponente), obtener la potencia del primero elevado al segundo sin utilizar el método “pow”. Por ejemplo, base = 3 y exponente = 5, $3^5 = 3 \times 3 \times 3 \times 3 \times 3 = 243$. Casos particulares: $0^0 = \text{Error}$ y $n^0 = 1$. (BaseExponente)

42. Mostrar todos los números pares comprendidos entre dos dados. (ParesEntreDosDados)

43. Dado un entero no negativo, se mostrarán todos los divisores de dicho entero separados por una coma. (Divisores)

44. Dado un entero no negativo se comprobará si es primo. Un número es primo si sus únicos divisores son el 1 y el mismo. (Primo)

45. Simúlese el lanzamiento de un dado. Se introducirá el número de veces que se lanza el dado (de no ser positivo, se establecerá a 100). (Dado)

NOTA En los ejercicios anteriores si los datos introducidos no cumplieran el rango considerado, se les asignará como valor el menor de dicho rango

Contadores

46. Introducir cinco enteros, y contabilizar el número de ceros introducidos. (ContarCeros)

47. Introducir cinco enteros, y comprobar si hay más positivos que negativos, más negativos que positivos o si hay el mismo número de positivos y negativos (el 0 no es positivo ni negativo). (ContarPositivosNegativos)

Señales

48. Introdúzcase un máximo de 5 enteros y compruébese a medida que se van introduciendo que los signos de tales valores se van alternando (el 0 no es positivo ni

negativo). La lectura de datos se terminará en cuanto que no se cumpla la alternancia de signos para el último valor leído. (AlternarSignos)

- 49.** Considérese un número de 5 dígitos. Además, se introducirá las veces que se puede repetir un dígito (de 1 a 5) y el dígito (de 0 al 9). Muéstrese la posición (unidades, decenas, centenas, ...) del dígito introducido que se repite la primera, segunda o tercera vez introducida contando desde las unidades. Por ejemplo, el dígito 1 aparece por segunda vez en el número 71871 en las unidades de mil. En el caso de que el dígito en cuestión no se repitiera las veces indicadas en el número, se pondría como solución “No aparece el dígito las veces indicadas”. (RepeticionesDigito)

Acumuladores

- 50.** Introducir cuatro enteros y obtener la suma de los impares (SumarImpares)
- 51.** Se introducirán las alturas de 8 personas. Se leerá una altura “a”. Se obtendrá la altura media redondeada a dos decimales de las personas que miden menos de “a” y de las que miden más o igual que “a”. (AlturasMedias). (AlturasMedias)

Salto

- 52.** Hágase un programa que lea enteros entre -10 y 10 hasta introducir el 0. De introducirse un valor fuera de dicho rango, se lanzará un mensaje y se volverá a leer un nuevo entero. Para los valores válidos introducidos, se contará el número de negativos y positivos. (SaltoContinue)

Ciclos

- 53.** Generar números enteros aleatorios entre 0 y 100 hasta obtener el 0. El programa mostrará cada uno de los datos generados y obtendrá el mayor de los números generados. (MayorNumero)
- 54.** Introducir un entero no negativo, y convertirlo a binario, octal y hexadecimal. (BinarioOctalHexadecimal)
- 55.** Leer un entero no negativo y una base entre 2 y 30. El programa obtendrá el número leído en la base dada. (ConvertirBase)

Anidamiento de bucles

- 56.** Dada la fecha de nacimiento de una persona (ddmmaaaa) se obtiene como número de la suerte un número entre 0 y 9 que se obtiene sumando los dígitos de su fecha de nacimiento; si el número obtenido es mayor que 9, se vuelven a sumar los dígitos de dicho número, y así sucesivamente, hasta que el número obtenido sea menor o igual que 9. Por ejemplo, el número de la suerte del 25022016 sería el 9, a saber: $2+5+0+2+2+0+1+6=18 \rightarrow 1+8=9$. (NumeroDeLaSuerte)

57. Dado un número positivo, se mostrarán todos los números primos menores que el valor introducido. (PrimosMenores)
58. Dado un número positivo (por ejemplo n), se mostrarán los “n” primeros números primos. (EnePrimos)
59. Representése los siguientes rectángulos y cuadrados. Para los rectángulos se introduce el ancho (mínimo 2) y el alto (mínimo 2), y para los cuadrados el lado (mínimo 2). De ser la medida introducida menor que 2 o no válida, se establecerá al primer valor válido.

a)

*	*	*	*	*	*
*	+	+	+	+	*
*	+	+	+	+	*
*	*	*	*	*	*

b)

*	+	*	+	*	+
+	*	+	*	+	*
*	+	*	+	*	+
+	*	+	*	+	*

c) De lado un número impar

		*		
		*		
*	*	*	*	*
		*		
		*		

d) De lado un número impar

*				
+	+			
+	a	+		
+	a	a	+	
*	+	+	+	*

e) De lado un número impar

*				*
	*		*	
		*		
	*		*	
*				*

f) De lado un número impar

		*		
	*	+	*	
*	+	+	+	*
	*	+	*	
		*		

g) De lado un número impar

*	*	*		
*		*		
*	*	*	*	*
		*		*
		*	*	*

(DibujoA, DibujoB, DibujoC, DibujoD, DibujoE, DibujoF, DibujoG)

60. Genérese un tablero cuyo ancho y alto se introduce por consola (mínimo 2). Las celdas tendrán un 0 o un 1 generado de forma aleatoria.

```

ANCHO: 5
ALTO: 5
|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|

```

(TableroAleatorio)

Tabla de ejecución con funciones

61. Para el siguiente programa se realizará la tabla de ejecución para los siguientes datos:
120.092 1 67087.65 -3 ¿Qué hace la función?

```

import fsg.in;
public class Funciones
{
    public static void main(String[] args)
    {
        double a = 0, r = 0;
        int n = 0;
    }
}

```

```

        for(int i=0;i<2;i++)
        {
            a = in.leerDouble("Real: ",v->v>0);
            n = in.leerInt();
            r = redondear(a,n);
            System.out.println(r);
        }
    }

    static double redondear(double v, int n)
    {
        double pot = 0;

        pot = Math.pow(10,n);
        v = v * pot;
        v = (int) (v+0.5);
        v = v / pot;

        return v;
    }
}

```

Ejemplo función

62. Hágase la tabla de ejecución para `coste(1.784, 34.5)`. ¿Qué hace la función?

```

static double coste(double precioLitro, double litros)
{
    double coste = 0;

    coste = precioLitro*litros;
    coste = (int) (coste*100) / 100.0;

    return coste;
}

```

Funciones

Hágase el diagrama de flujo y el código de las siguientes aplicaciones:

63. Genérese 10 números aleatorios para cada uno de los cuales se introduce el desde y el hasta. Para ello se definirá una función que dado dos enteros devuelva un número aleatorio entre los dos (ambos incluidos). (MetodoAleatorio)

64. Dado un número natural (mayor o igual que 0) se comprobará si es capicúa. Para ello se definirán e invocarán los siguientes métodos:

- Una función que dado un entero devuelva su inverso.
- Una función que dado un entero devuelva si es o no capicúa.

(MetodoCapicua)

65. Se quiere realizar el cambio de determinadas cantidades en euro a dólares. Para cada cantidad a convertir (se validará que de ser positiva no tenga más de 2 decimales), se introducirá el cambio de 1 euro a dólares (se validará que dicho valor sea mayor que 0). El usuario introducirá cantidades en euro a convertir hasta introducir un valor 0 o negativo. Se tendrán los siguientes métodos que serán invocados:

- Una función que lea la cantidad en euros valida a convertir y lo devuelva.
- Una función que lea el cambio válido y lo devuelva.
- Una función que dada la cantidad en euro y el cambio de euro a dolar devuelva el valor en dólares truncado a dos decimales.
- Una función que dada la cantidad de euros a convertir, el cambio de euro a dólar devuelva una cadena con el siguiente formato:

EUROS: XXXX EURO/DÓLAR: XXXX CAMBIO: XXXX

NOTA: Primero se leerán todos los datos y al final se mostrarán los cambios.

(MetodoCambio)

Procedimientos

66. Léanse el peso y la altura de 10 personas (se validará que el peso es un valor entre 2.00 y 120.00 y la altura entre 0.2 y 2.2). Para cada persona se obtendrá su IMC (índice de masa corporal) y su interpretación. Para ello se definirán e invocarán los siguientes métodos.

- Una función sin parámetros para leer el peso y lo devuelva.
- Una función sin parámetros para leer la altura y lo devuelva.
- Una función que dado el peso y la altura (parámetros) obtenga el IMC.
- Una función que dado el IMC (parámetro) devuelva la interpretación del índice de masa corporal.
- Un procedimiento que dados el peso, altura, el IMC y su interpretación muestra la siguiente salida por consola:

(se limpiará la consola)

INFORME DE INDICE DE MASA CORPORAL

PERSONA NUMERO: XX

```

PESO: XXXXX
ALTURA: XXXXX
IMC: XXXXX
INTERPRETACION: XXXXXXXXX
*****
(se detendrá el control durante 5 segundos)

```

NOTA El índice de masa corporal se obtiene como el cociente entre el peso en kilogramos y el cuadrado de la altura en metros redondeado a 1 decimal.

(MetodoIMC)

Tipos básicos y de cadenas por referencia

67. Hágase un programa que lea un número real y un entero. El programa duplicará el valor real leído, incrementará el nuevo valor real en la cantidad entera introducida y lo redondeará a las centésimas. Se definirán las siguientes funciones:

- Una función que duplique un valor real
- Una función que incremente un valor real en una determinada cantidad entera
- Una función que redondee a las centésimas un valor real

Método recursivo

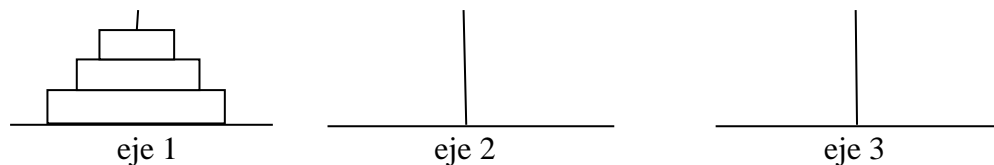
68. Se define la sucesión de Fibonacci de la siguiente manera:

$$F(0)=F(1)=1;$$

$$F(n)=F(n-1)+F(n-2) \text{ si } n>1$$

Léase un entero no negativo y obténgase el valor de dicha sucesión para el valor leído. Hágase de forma recursiva (FibRecursivo) e iterativa (FibIterativa).

69. Resuélvase el problema de la Torres de Hanoi para un número de discos entre 3 y 10. (TorresDeHanoi)



El problema de las Torres de Hanoi para 3 discos consiste en mover los tres discos del eje 1 al eje 3 con las siguientes reglas:

- En cada movimiento sólo se puede mover un disco
- Sólo se puede mover el disco de la parte superior de un eje
- El disco a mover sólo puede ir a un eje que no tenga discos o su disco superior tenga una base mayor que la del disco que se está moviendo

Menús

70. Hágase un menú que permita sumar, restar, multiplicar y dividir (división entera) dos números enteros que el usuario introducirá. (MenuOperar)

Inicio del menu

OPCION ACCION

```
=====
1   Sumar dos enteros
2   Restar dos enteros
3   Multiplicar dos enteros
4   Dividir dos enteros
5   Terminar
```

OPCION:

Elección de la opción 1

```
* ***** *
* SUMAR DOS ENTEROS *
* ***** *
Escribe un entero: 67
Escribe un entero: 11
La suma vale 78
```

Pulsa ENTER para continuar

Elección de la opción 2

```
* ***** *
* RESTAR DOS ENTEROS *
* ***** *
Escribe un entero: 34
Escribe un entero: 22
La resta vale 12
```

Pulsa ENTER para continuar

Elección de la opción 3

```
* ***** *
* MULTIPLICAR DOS ENTEROS *
* ***** *
Escribe un entero: 78
Escribe un entero: 23
El producto vale 1794
```

Pulsa ENTER para continuar

Elección opción 4

```

* ***** *
* DIVIDIR DOS ENTEROS *
* ***** *
Escribe un entero: 89
Escribe un entero: 12
La division vale 7

Pulsa ENTER para continuar

* ***** *
* DIVIDIR DOS ENTEROS *
* ***** *
Escribe un entero: 34
Escribe un entero: 0
No se puede realizar la división

Pulsa ENTER para continuar

Elección opción 5
OPCION ACCION
=====
1 Sumar dos enteros
2 Restar dos enteros
3 Multiplicar dos enteros
4 Dividir dos enteros
5 Terminar

OPCION: 5
FIN DEL PROGRAMA

```

71. Hágase un menú que permita realizar lo siguiente:

- Dibujar un cuadrado (se leerá un lado impar, y se mostrará un aspa),
- Mostrar un informe del coste de las consumiciones para una serie de clientes (se leerán el número de clientes, y para cada cliente se indicarán cuántas consumiciones han realizado y el coste de cada consumición – el mismo para cada consumición-). El número de consumiciones será un valor entre 1 y 10, como máximo. El coste de las consumiciones serán un valor entre 1 y 2,5 y como máximo tendrán dos decimales.
- Leer un número positivo y comprobar si es un número de Armstrong (un número de “n” dígitos es de Armstrong si la suma de sus dígitos elevado cada uno de ellos a “n” es el número en cuestión).
- Operar dos enteros. Al seleccionar esta opción se mostrará el menú de la práctica anterior.
(MenuConSubmenu)

Instrucción try-catch

72. Hágase un programa que genere de manera aleatoria 100 valores enteros entre -5 y 5, ambos incluidos. Para cada par de datos se mostrará su división (por ejemplo, para los datos 7 y 2 se mostraría “7/2 = 3”). Identifíquese la posible excepción y

modifíquese para que cuando se provoque dicha excepción se muestre por consola el mensaje “No se puede dividir”

Librerías de usuario

73. Defínase o añádase a la librería “lib” los siguientes métodos:

- Un método “aleatorio” para generar números aleatorios entre dos enteros, ambos inclusive.
- Un método “aleatorio” sobrecargado para generar números aleatorios entre dos reales, el último sin incluir.
- Un método “truncar” para truncar enteros y reales a una posición determinada.
- Un método “redondear” para redondear enteros y reales a una posición determinada.

Hágase una aplicación que genere un número real aleatorio entre 10 y 100 (este último sin incluir), lo trunque a las decenas y lo redondee a las milésimas. Se mostrarán por consola los tres valores. Utilícese los métodos definidos en la librería. (LibreriaUsuario)

Validación de datos de entrada de cadena

74. Leer una palabra que sea un mes del año en mayúsculas. (FormatoMes)

75. Se introduce una cadena y se comprueba que es una palabra que comienza por mayúsculas y el resto en minúsculas. (FormatoPalabra)

76. Léase una línea y compruébese que sólo aparece el carácter ‘1’ exactamente tres veces. (UnoTresVeces)

77. Léase una línea y compruébese si aparece al menos tres consonantes en minúsculas seguidas. (TresConsonantesSeguidas)

Las clase String y StringBuffer

78. Leer un nombre y un apellido y separarlo por un solo espacio.



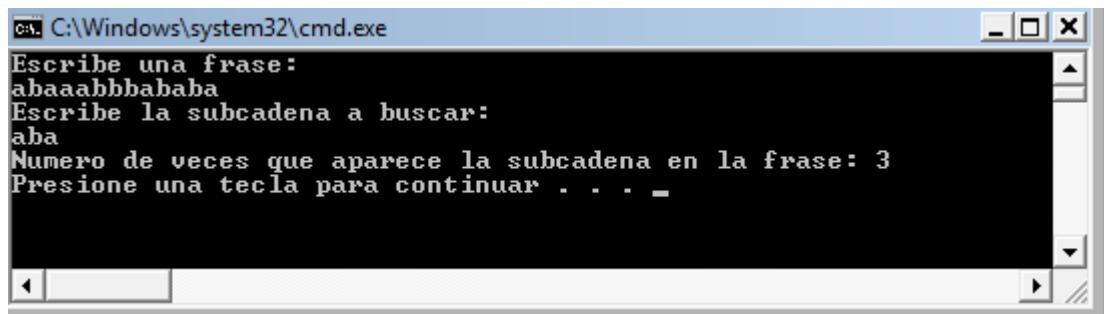
(Separar.java)

79. Léase una frase, póngase en mayúsculas, inviértase la cadena, y muéstrese.



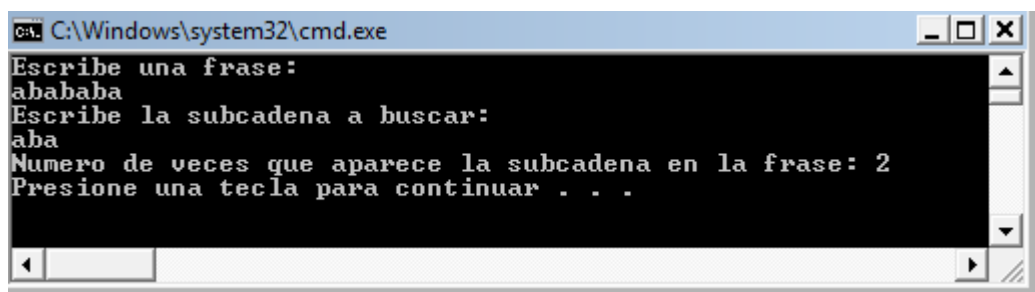
(MetodosCadena)

80. Contar las veces que una cadena aparece como subcadena en otra cadena con solapamiento.



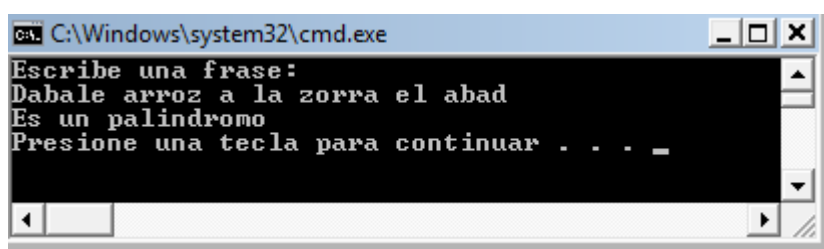
(Subcadena.java)

81. Contar las veces que una cadena aparece como subcadena en otra cadena sin solapamiento.



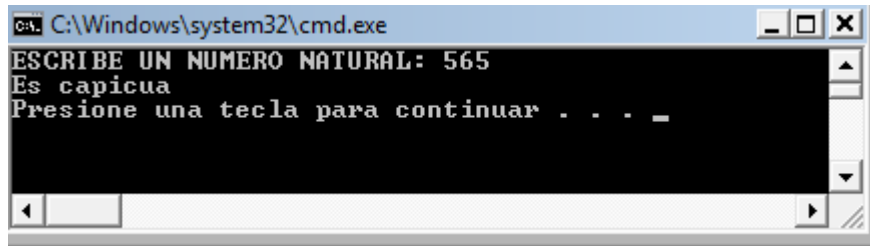
(SubcadenaSin.java)

82. Comprobar si una frase es un palíndromo.



(Palíndromo.java)

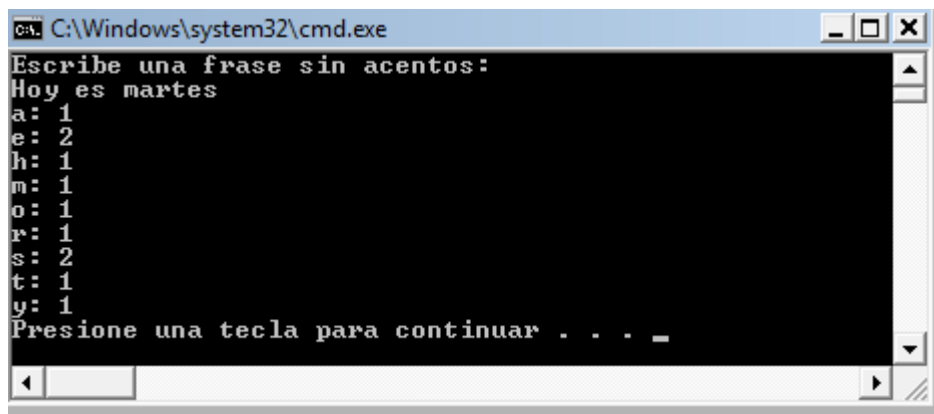
83. Comprobar si un número es capicúa.



```
C:\Windows\system32\cmd.exe
ESCRIBE UN NUMERO NATURAL: 565
Es capicua
Presione una tecla para continuar . . . _
```

(CapicuaString.java)

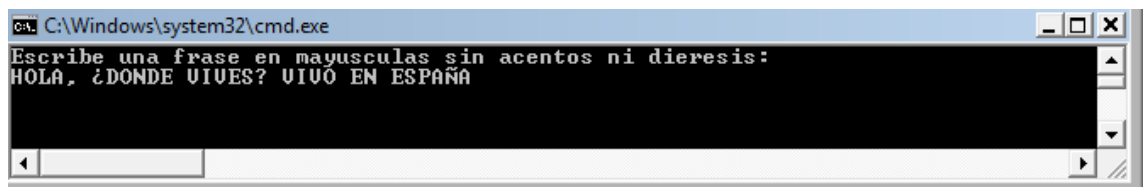
84. Contar el número de letras de las A a la Z, ambas inclusive, introducidas en una frase.



```
C:\Windows\system32\cmd.exe
Escribe una frase sin acentos:
Hoy es martes
a: 1
e: 2
h: 1
m: 1
o: 1
r: 1
s: 2
t: 1
y: 1
Presione una tecla para continuar . . . _
```

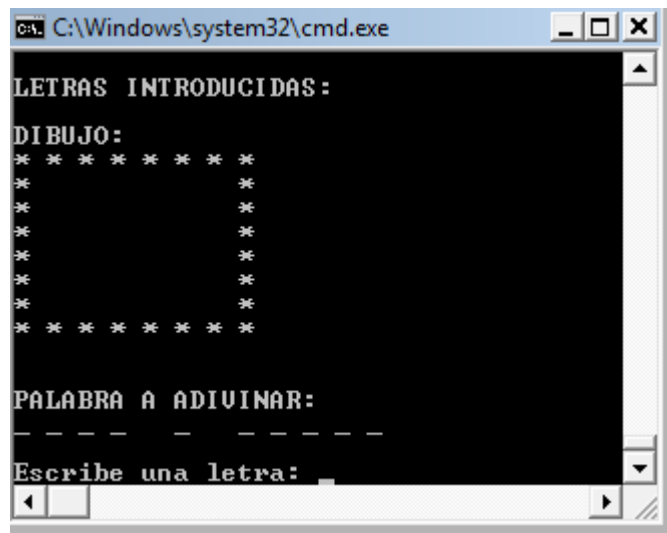
(ContarLetras.java)

85. Simúlese el juego del ahorcado. El usuario introducirá la frase a adivinar. Las palabras estarán en mayúsculas, sin acentos y separadas por un espacio. El otro jugador tendrá 12 oportunidades para adivinar la frase, introduciendo cada vez una letra.



```
C:\Windows\system32\cmd.exe
Escribe una frase en mayusculas sin acentos ni dieresis:
HOLA, ¿DONDE VIVES? VIUÓ EN ESPAÑA
```

Pantalla donde se introduce la frase a adivinar



El jugador introduce letras en mayúsculas.