**Trabajo práctico nro. 1**

|  |  |  |
| --- | --- | --- |
| Logo Departamento 2.jpg | **Asignatura: Laboratorio I** | |
|  | |
| **Cursado:**Primer Trimestre | **Horassemanales**: |
|  | **Horas semestrales:**  *Cantidad estimada de horas semestrales/anuales.* |
| **Carrera**: *Tecnicatura Universitaria en Programación* | **Nivel (Año):** |
| **Ciclo Lectivo: 2023** |

**Integrantes de la Cátedra:**

* **DOCENTES:**

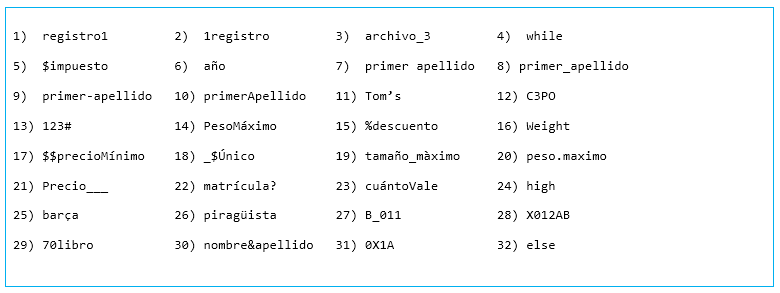
|  |  |  |
| --- | --- | --- |
| **Nombre del Profesor** | **Periodo** | **Cantidad horas materia** |
|  |  | 6 horas |

1. **Tema:**

**Ejercicios Básicos iniciales para principiantes**

1. **Enunciados:**
2. **Ejercicio sobre identificadores: indica qué identificadores Java son válidos.**

Indica cuáles de los siguientes identificadores son válidos en Java. Si el identificador no es válido explica porqué no lo es.

1) registro1: válido.

2) 1registro: No válido. Un identificador no puede comenzar por un dígito.

3) archivo\_3: válido.

4) while: No válido. Es una palabra reservada del lenguaje.

5) $impuesto: válido.

6) año: válido.

7) primer apellido: no válido, un identificador no puede contener espacios en blanco.

En todo caso serían dos identificadores.

8) primer\_apellido: válido.

9) primer-apellido: No válido. Un identificador no puede contener el carácter – y en general no puede contener

caracteres que representen operadores: +, -, \*, /, %, <, >, etc.

10) primerApellido: válido.

11) Tom’s: No válido. Un identificador no puede contener el carácter comilla simple

12) C3PO: válido.

13) 123#: No válido. El carácter # no es válido en un identificador.

14) PesoMáximo: válido. Las vocales acentuadas pueden usarse para crear identificadores.

15) %descuento: No válido. Contiene el carácter no válido %

16) Weight: válido

17) $$precioMínimo: válido

18) \_$Único: válido

19) tamaño\_màximo: válido

20) peso.maximo: No válido. Un identificador no puede contener el carácter ‘.’

21) Precio\_\_\_: válido

22) matrícula?: No válido. Un identificador no puede contener el carácter ‘?’

23) cuántoVale: válido

24) high: válido

25) barça: válido

26) piragüista: válido

27) B\_011: válido

28) X012AB: válido

29) 70libro: No válido. Un identificador no puede comenzar por un dígito.

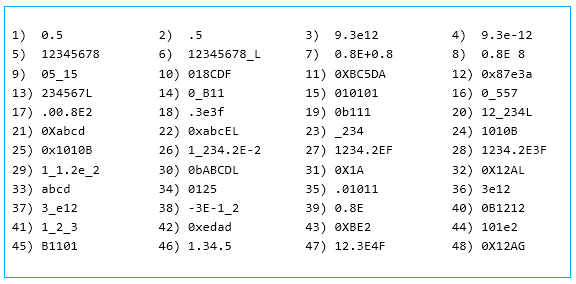
30) nombre&apellido: No válido. Un identificador no puede contener el carácter ‘&’

31) 0X1A: No válido. Un identificador no puede comenzar por un dígito.

32) else: No válido. Es una palabra reservada del lenguaje.

1. **Ejercicio sobre literales: indica qué literales Java son válidos**.

Indica cuáles de los siguientes literales son válidos en Java. Si el literal es válido indica además de qué tipo es (int, double, long, etc) y el sistema de numeración en el que está escrito (decimal, binario, octal, hexadecimal). Si el literal no es válido explica porqué no lo es.

LITERAL VÁLIDO TIPO SISTEMA NUMERACIÓN

------------------------------------------------------------------------------------------

1) 0.5 SI double decimal

------------------------------------------------------------------------------------------

2) .5 SI. double decimal

Si la parte entera es 0 se puede omitir

------------------------------------------------------------------------------------------

3) 9.3e12 SI. double decimal

Representa el valor 9.3\*1012

------------------------------------------------------------------------------------------

4) 9.3e-12 SI. double decimal

Representa el valor 9.3\*10-12

------------------------------------------------------------------------------------------

5) 12345678 SI int decimal

------------------------------------------------------------------------------------------

6) 12345678\_L NO. Posición no válida para el carácter \_

------------------------------------------------------------------------------------------

7) 0.8E+0.8 NO. Valor no válido para el exponente en un literal Java.

No puede contener decimales. Intenta representar el valor 0.8\*100.8

------------------------------------------------------------------------------------------

8) 0.8E 8 NO. Hay un espacio en blanco entre E y 8

------------------------------------------------------------------------------------------

9) 05\_15 SI int octal

------------------------------------------------------------------------------------------

10) 018CDF NO. Si el número comienza por 0 está indicando que es un entero

escrito en octal pero contiene los caracteres CDF no válidos para el

sistema octal. Si fuese un entero hexadecimal debería comenzar por 0X.

------------------------------------------------------------------------------------------

11) 0XBC5DA SI int hexadecimal

------------------------------------------------------------------------------------------

12) 0x87e3a SI int hexadecimal

------------------------------------------------------------------------------------------

13) 234567L SI long decimal

------------------------------------------------------------------------------------------

14) 0\_B11 NO. El carácter \_ no puede separar los caracteres 0B que indican que

el número está escrito en binario.

------------------------------------------------------------------------------------------

15) 010101 SI int octal

------------------------------------------------------------------------------------------

16) 0\_557 SI int octal

------------------------------------------------------------------------------------------

17) .00.8E2 NO. El literal solo puede contener un punto (.) que separe la parte

entera de la parte decimal.

------------------------------------------------------------------------------------------

18) .3e3f SI float decimal

------------------------------------------------------------------------------------------

19) 0b111 SI int binario

------------------------------------------------------------------------------------------

20) 12\_234L SI long decimal

------------------------------------------------------------------------------------------

21) 0Xabcd SI int hexadecimal

------------------------------------------------------------------------------------------

22) 0xabcEL SI long hexadecimal

------------------------------------------------------------------------------------------

23) \_234 NO. El carácter \_ no puede aparecer al principio del número.

------------------------------------------------------------------------------------------

24) 1010B NO. Si fuese binario debería empezar por 0B: 0B1010

------------------------------------------------------------------------------------------

25) 0x1010B SI int hexadecimal

------------------------------------------------------------------------------------------

26) 1\_234.2E-2 SI double decimal

Representa el valor 1234.2\*10-2

------------------------------------------------------------------------------------------

27) 1234.2EF NO. Falta el valor del exponente entre los caracteres E y F

------------------------------------------------------------------------------------------

28) 1234.2E3F SI float decimal

Representa el valor 1234.2\*103

------------------------------------------------------------------------------------------

29) 1\_1.2e\_2 NO. En un literal Java de tipo float o double el carácter \_ no puede

aparecer antes o después del carácter e.

En este caso se quiere representar el valor 11.2\*102

------------------------------------------------------------------------------------------

30) 0bABCDL NO. Comienza por 0b. Si fuese hexadecimal debería comenzar por 0X.

------------------------------------------------------------------------------------------

31) 0X1A SI int hexadecimal

------------------------------------------------------------------------------------------

32) 0X12AL SI long hexadecimal

------------------------------------------------------------------------------------------

33) abcd NO. Si fuese hexadecimal debería comenzar por 0X.

------------------------------------------------------------------------------------------

34) 0125 SI int octal

------------------------------------------------------------------------------------------

35) .01011 SI double decimal

------------------------------------------------------------------------------------------

36) 3e12 SI double decimal

Representa el valor 3\*1012

------------------------------------------------------------------------------------------

37) 3\_e12 NO. En un literal Java de tipo float o double el carácter \_ no puede

aparecer antes o después del carácter e.

En este caso se quiere representar el valor 3\*1012

------------------------------------------------------------------------------------------

38) -3E-1\_2 SI double decimal

Representa el valor -3\*10-12

------------------------------------------------------------------------------------------

39) 0.8E NO. Falta el valor del exponente

------------------------------------------------------------------------------------------

40) 0B1212 NO. Un número binario (0B) solo puede contener ceros y unos

------------------------------------------------------------------------------------------

41) 1\_2\_3 SI int decimal

------------------------------------------------------------------------------------------

42) 0xedad SI int hexadecimal

------------------------------------------------------------------------------------------

43) 0XBE2 SI int hexadecimal

------------------------------------------------------------------------------------------

44) 101e2 SI double decimal

Representa el valor 101\*102

------------------------------------------------------------------------------------------

45) B1101 NO. Si fuese binario debería empezar por 0B

------------------------------------------------------------------------------------------

46) 1.34.5 NO. El literal solo puede contener un punto (.) que separe la parte

entera de la parte decimal.

------------------------------------------------------------------------------------------

47) 12.3E4F SI float decimal

Representa el valor 12.3\*104

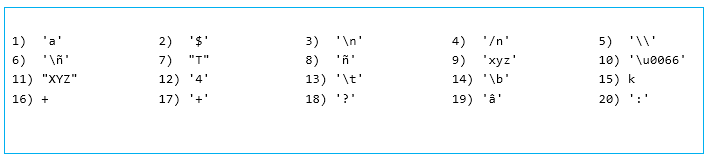
------------------------------------------------------------------------------------------

48) 0X12AG NO. G no es un carácter válido para un número hexadecimal (0X)

------------------------------------------------------------------------------------------

1. **Ejercicio sobre literales de tipo char.**

Indica cuáles de los siguientes literales de tipo char son válidos en Java. Si el literal no es válido explica el motivo.



LITERAL VÁLIDO

------------------------------------------------------------------------------------------

1) 'a' SI

------------------------------------------------------------------------------------------

2) '$' SI

------------------------------------------------------------------------------------------

3) '\n' SI Es una secuencia de escape

------------------------------------------------------------------------------------------

4) '/n' NO No válido. Un literal de tipo carácter debe estar formado por un

solo carácter a no ser que sea una secuencia de escape.

------------------------------------------------------------------------------------------

5) '\\' SI Es una secuencia de escape

------------------------------------------------------------------------------------------

6) '\ñ' NO Un literal de tipo carácter debe estar formado por un solo

carácter a no ser que sea una secuencia de escape.

------------------------------------------------------------------------------------------

7) "T" NO Los literales de tipo carácter deben ir entre comillas simples.

------------------------------------------------------------------------------------------

8) 'ñ' SI

------------------------------------------------------------------------------------------

9) 'xyz' NO Un literal de tipo carácter debe estar formado por un solo

carácter a no ser que sea una secuencia de escape.

------------------------------------------------------------------------------------------

10) '\u0066' SI Representa un valor Unicode.

------------------------------------------------------------------------------------------

11) "XYZ" NO Los literales de tipo carácter están formados por un solo

carácter y deben ir entre comillas simples.

------------------------------------------------------------------------------------------

12) '4' SI

------------------------------------------------------------------------------------------

13) '\t' SI Es una secuencia de escape

------------------------------------------------------------------------------------------

14) '\b' SI Es una secuencia de escape

------------------------------------------------------------------------------------------

15) k NO Los literales de tipo char deben inr entre comillas simples.

------------------------------------------------------------------------------------------

16) + NO Los literales de tipo char deben inr entre comillas simples.

------------------------------------------------------------------------------------------

17) '+' SI

------------------------------------------------------------------------------------------

18) '?' SI

------------------------------------------------------------------------------------------

19) 'â' SI

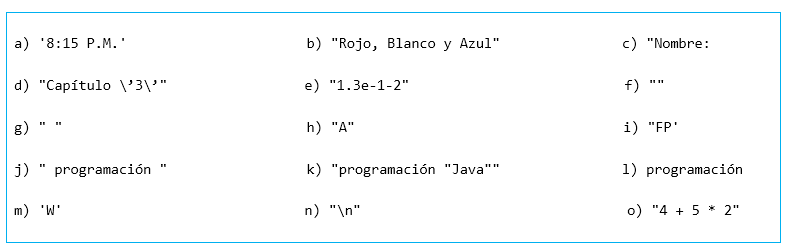
------------------------------------------------------------------------------------------

20) ':' SI

------------------------------------------------------------------------------------------

1. **Ejercicio sobre literales de tipo String.**

Indica cuáles de los siguientes literales de tipo String son válidos en Java. Si el literal no es válido explica el motivo.



LITERAL VÁLIDO

------------------------------------------------------------------------------------------

a) '8:15 P.M.' NO Los literales de tipo String deben ir entre

comillas dobles

------------------------------------------------------------------------------------------

b) "Rojo, Blanco y Azul" SI

------------------------------------------------------------------------------------------

c) "Nombre: NO Faltan las comillas finales

------------------------------------------------------------------------------------------

d) "Capítulo \’3\’" SI Un literal de tipo String puede contener secuencias

de escape. En este caso contiene la secuencia \'

------------------------------------------------------------------------------------------

e) "1.3e-1-2" SI

------------------------------------------------------------------------------------------

f) "" SI Representa una cadena vacía.

------------------------------------------------------------------------------------------

g) " " SI Es un String formado por un solo espacio en blanco

------------------------------------------------------------------------------------------

h) "A" SI

------------------------------------------------------------------------------------------

i) "FP' NO Las comillas finales en un String deben ser dobles.

------------------------------------------------------------------------------------------

j) " programación " SI

------------------------------------------------------------------------------------------

k) "programación "Java"" NO Para que fuese válido, el String debería contener las

comillas interiores como secuencias de escape

"programación \"Java\""

------------------------------------------------------------------------------------------

l) programación NO Los literales de tipo String deben en ir entre

comillas dobles

------------------------------------------------------------------------------------------

m) 'W' NO Los literales de tipo String deben en ir entre

comillas dobles

------------------------------------------------------------------------------------------

n) "\n" SI

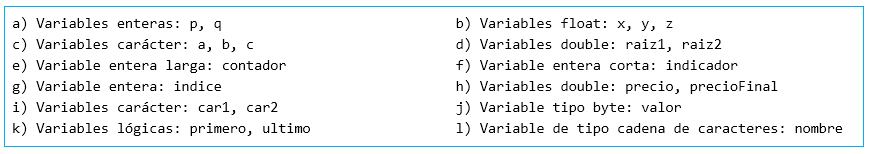
------------------------------------------------------------------------------------------

o) "4 + 5 \* 2" SI

------------------------------------------------------------------------------------------

1. Ejercicio sobre declaraciones de variables

**Ejercicio 1**: Escribe la declaración para cada una de las variables y asígnales un valor inicial en la propia declaración de variable.



a) Variables enteras: p, q int p = 0, q = 3;

b) Variables float: x, y, z float x = 2.5F, y = 23.005F, z = 1F;

c) Variables carácter: a, b, c char a = 'g', b = 'k', c = 'q';

d) Variables double: raiz1, raiz2 double raiz1 = 4.55, raiz2 = 10.25;

e) Variable entera larga: contador long contador = 0;

f) Variable entera corta: indicador short indicador = 30;

g) Variable entera: indice int indice = 276;

h) Variables double: precio, precioFinal double precio = 30, precioFinal = 29.99;

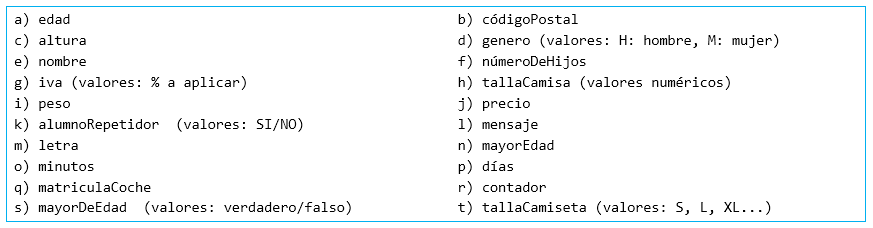
i) Variables carácter: car1, car2 char car1 = 'a', car2 = 'b';

j) Variable tipo byte: valor byte valor = 4;

k) Variables lógicas: primero, ultimo boolean primero = true, ultimo = false;

l) Variable de tipo cadena de caracteres: nombre String nombre = "Antonio Abad";

**Ejercicio 2**: Escribe la declaración más apropiada para cada una de las siguientes variables. El nombre de cada una indica el tipo de dato que contendrá y servirá para determinar el tipo de dato más adecuado en la declaración de la variable.



a) edad int edad;

b) códigoPostal String códigoPostal;

c) altura double altura;

d) genero (H: hombre, M: mujer) char genero;

e) nombre String nombre;

f) númeroDeHijos int númeroDeHijos;

g) iva (%) double iva;

h) tallaCamisa (valores numéricos) int tallaCamisa;

i) peso double peso;

j) precio double precio;

k) alumnoRepetidor (valores: SI/NO) String alumnoRepetidor;

l) mensaje String mensaje;

m) letra char letra;

n) mayorEdad int mayorEdad;

o) minutos int minutos;

p) días int días;

q) matriculaCoche String matrículaCoche;

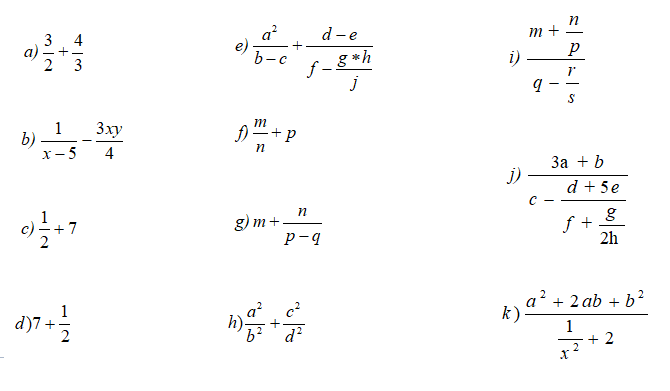
r) contador int contador

s) mayorDeEdad (valores: verdadero/falso) boolean mayorDeEdad;

t) tallaCamiseta (valores: S, L, XL...) String tallaCamiseta;

6. Convertir expresiones algebráicas en expresiones algorítmicas

**Ejercicio:** Convierte en expresiones algorítmicas las siguientes expresiones algebráicas. Coloca paréntesis solamente donde sean necesarios.



a) 3/2+4/3

e) a\*a/(b-c)+(d-e)/(f-g\*h/j)

i) (m+n/p)/(q-r/s)

b) 1/(x-5)-3\*x\*y/4

f) m/n+p

j) (3\*a+b)/(c-(d+5\*e)/(f+g/(2\*h)))

c) 1/2+7

g) m+n/(p-q)

k) (a\*a+2\*a\*b+b\*b)/(1/(x\*x))+2

d) 7+1/2

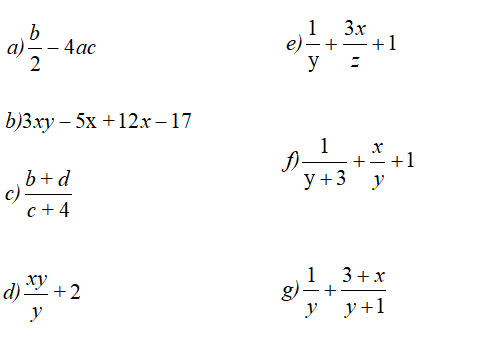
h) a\*a/(b\*b)+c\*c/(d\*d)

7. Convertir expresiones algorítmicas en expresiones algebráicas

**Ejercicio:** Convierte en expresiones algebráicas las siguientes expresiones algorítmicas.

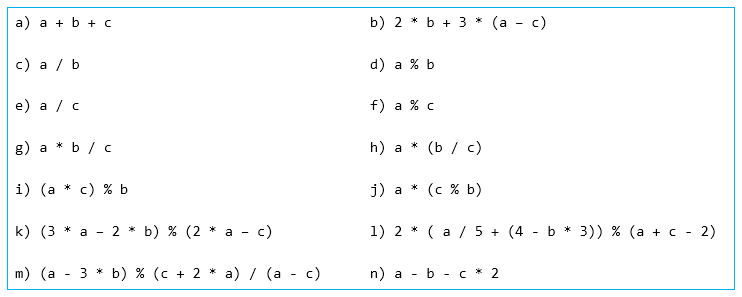
**Ejercicio 1:** Si a, b y c son variables enteras con valores a=8, b=3, c=-5, determina el valor de las siguientes expresiones aritméticas:





8. Ejercicio sobre operadores aritméticos

**Ejercicio 1:** Si a, b y c son variables enteras con valores a=8, b=3, c=-5, determina el valor de las siguientes expresiones aritméticas:



a) a + b + c --> 6

b) 2 \* b + 3 \* (a – c) --> 45

c) a / b --> 2

d) a % b --> 2

e) a / c --> -1

f) a % c --> 3

g) a \* b / c --> -4

h) a \* (b / c) --> 0

i) (a \* c) % b --> -1

j) a \* (c % b) --> -16

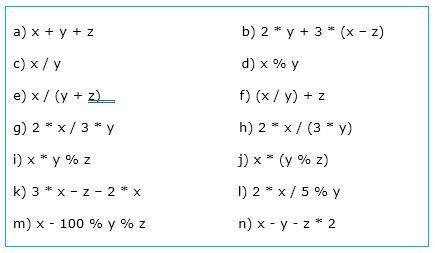
k) (3 \* a – 2 \* b) % (2 \* a – c) --> 18

l) 2 \* ( a / 5 + (4 - b \* 3)) % (a + c - 2) --> 0

m) (a - 3 \* b) % (c + 2 \* a) / (a - c) --> 0

n) a - b - c \* 2 --> 15

**Ejercicio 2:**Si x, y, z son variables de tipo double con valores x= 88, y = 3.5, z = -5.2, determina el valor de las siguientes expresiones aritméticas. Obtén el resultado de cada expresión con un máximo de cuatro decimales.



a) x + y + z --> 86.3

b) 2 \* y + 3 \* (x – z) --> 286.6

c) x / y --> 25.1428

d) x % y --> 0.5

e) x / (y + z) --> -51.7647

f) (x / y) + z --> 19.9428

g) 2 \* x / 3 \* y --> 205.3333

h) 2 \* x / (3 \* y) --> 16.7619

i) x \* y % z --> 1.1999

j) x \* (y % z) --> 308.0

k) 3 \* x – z – 2 \* x --> 93.1999

l) 2 \* x / 5 % y --> 0.2000

m) x - 100 % y % z --> 86.0

n) x - y - z \* 2 --> 94.9

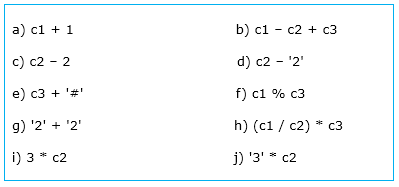
**Ejercicio 3.**Si c1, c2 y c3 son variables de tipo char con valores c1=’E’, c2=’5’, c3=’?’, determina el valor numérico de las siguientes expresiones aritméticas. Para resolverlo necesitas saber el valor numérico correspondiente a esos caracteres según la tabla ASCII:

’E’      69

’5’      53

’?’      63

Consulta en la tabla ASCII los valores numéricos del resto de caracteres que aparecen en las operaciones.



a) c1 + 1 --> 70

b) c1 – c2 + c3 --> 79

c) c2 – 2 --> 51

d) c2 – '2' --> 3

e) c3 + '#' --> 98

f) c1 % c3 --> 6

g) '2' + '2' --> 100

h) (c1 / c2) \* c3 --> 63

i) 3 \* c2 --> 159

j) '3' \* c2 --> 2703

9. Determinar el tipo del resultado en operaciones aritméticas

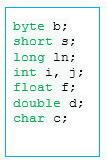
En aquellas expresiones aritméticas en las que intervienen operandos de distinto tipo (int, double, char, etc.) java convierte el tipo de los operandos que intervienen al tipo del operando de mayor precisión y este será el tipo del resultado obtenido en la operación.

Esta conversión de tipos se realiza de forma temporal, solamente para el cálculo de la operación. Los tipos originales de los operandos que intervienen siguen siendo los mismos después de la operación.

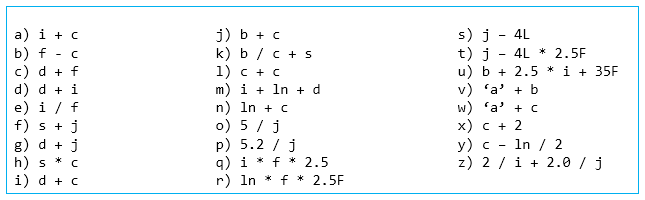
Debemos tener en cuenta que cuando se realiza la conversión temporal de tipos, los operandos de tipo short, byte y char se convierten a int.

Teniendo en cuenta todo esto, realiza el siguiente ejercicio.

**Ejercicio**: A partir de las siguientes declaraciones de variables:



Determina cuál es el tipo de dato del resultado de las siguientes expresiones aritméticas:



a) i + c (int) j) b + c (int) s) j – 4L (long)

b) f – c (float) k) b / c + s (int) t) j – 4L \* 2.5F (float)

c) d + f (double) l) c + c (int) u) b + 2.5 \* i + 35F (double)

d) d + i (double) m) i + ln + d (double) v) ‘a’ + b (int)

e) i / f (float) n) ln + c (long) w) ‘a’ + c (int)

f) s + j (int) o) 5 / j (int) x) c + 2 (int)

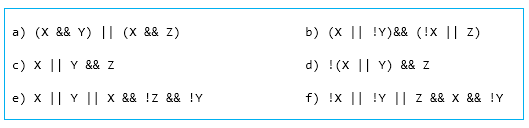
g) d + j (double) p) 5.2 / j (double) y) c – ln / 2 (int)

h) s \* c (int) q) i \* f \* 2.5 (double) z) 2 / i + 2.0 / j (double)

i) d + c (double) r) ln \* f \* 2.5F (float)

10. Ejercicio sobre operadores lógicos

**Ejercicio 1:** Si X, Y y Z son variables de tipo boolean con valores X = true, Y = false, Z = true, determina el valor de las siguientes expresiones lógicas:



a) (X && Y) || (X && Z) => true b) (X || !Y) && (!X || Z) => true

(T && F) || (T && T) (T || !F) && (!T || T)

F || T (T || T) && (F || T)

T T && T

T

c) X || Y && Z => true d) !(X || Y) && Z => false

T || F && T !(T || F) && T

T || F !T && T

T F && T

F

e) X || Y || X && !Z && !Y => true f) !X || !Y || Z && X && !Y => true

T || F || T && !T && !F !T || !F || T && T && !F

T || F || T && F && T F || T || T && T && T

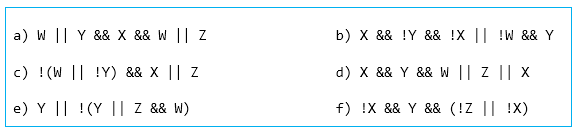
T || F || F && T F || T || T && T

T || F || F F || T || T

T || F T || T

T T

**Ejercicio 2:**Si W, X, Y y Z son variables de tipo boolean con valores W = false, X = true, Y = true, Z = false, determina el valor de las siguientes expresiones lógicas:



a) W || Y && X && W || Z => false b) X && !Y && !X || !W && Y => true

F || T && T && F || F T && !T && !T || !F && T

F || T && F || F T && F && F || T && T

F || F || F F && F || T && T

F || F F || T

F T

c) !(W || !Y) && X || Z => true d) X && Y && W || Z || X => true

!(F || !T) && T || F T && T && F || F || T

!(F || F) && T || F T && F || F || T

!F && T || F F || F || T

T && T || F F || T

T || F T

T

e) Y || !(Y || Z && W) => true f) !X && Y && (!Z || !X) => false

T || !(T || F && F) !T && T && (!F || !T)

T || !(T || F ) F && T && (T || F)

T || !( T ) F && T && T

T || F F && T

T F

Para resolver ejercicios con operadores lógicos tienes que conocer las tablas de verdad de estos operadores:

|  |  |  |
| --- | --- | --- |
| Operador AND | Operador OR | Operador NOT |
|  |  |  |

Como puedes observar:

* El operador lógico && (AND) da como resultado **true** cuando ambos operandos son true, en cualquier otro caso el resultado es false.
* El operador || (OR) da como resultado **false** cuando ambn cualquier otro caso el resultado es true.
* El operador ! (NOT) cambia el valor lógico del operando.

También debemos tener en cuenta para resolver estos ejercicios cuál es la precedencia de los operadores lógicos en Java:

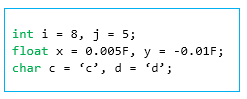
|  |
| --- |
| Precedencia de los operadores lógicos |
|  |

11. Ejercicios sobre operadores relacionales

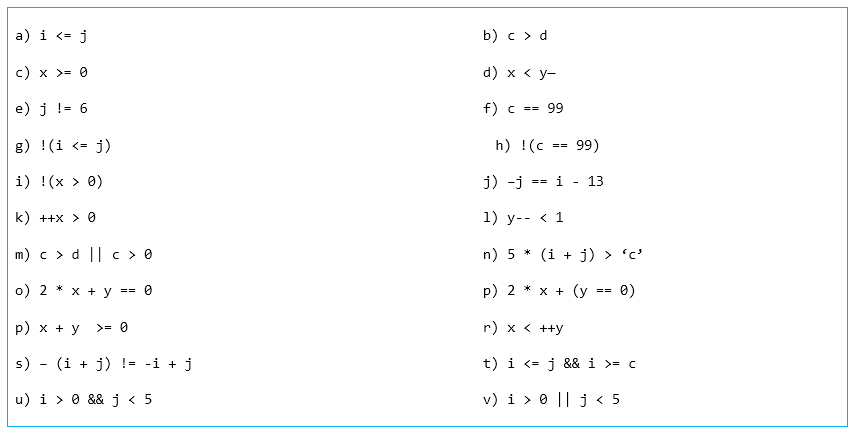
En estos ejercicios practicaremos con los operadores Java relacionales. En algunos de los ejercicios, los operadores relacionales aparecerán en expresiones en los que también intervienen el resto de operadores Java: operadores aritméticos, lógicos y los operadores incremento y decremento.

**Ejercicio:**

A partir de las siguientes declaraciones de variables:



Determina el valor de las siguientes expresiones:



a) i <= j

result = i <= j; // false

b) c > d

result = c > d; // false

c) x >= 0

result = x >= 0; // true

d) x < y--

result = x < y--; // false, y se decrementa después de la comparación

e) j != 6

result = j != 6; // true

f) c == 99

result = c == 99; // false, 'c' tiene un valor numérico diferente

g) !(i <= j)

result = !(i <= j); // true, la negación invierte el resultado

h) !(c == 99)

result = !(c == 99); // true, 'c' no es igual a 99

i) !(x > 0)

result = !(x > 0); // false, x es mayor que 0

j) -j == i - 13

result = -j == i - 13; // true, -5 es igual a 8 - 13

k) ++x > 0

result = ++x > 0; // true, después del incremento, x es positivo

l) y-- < 1

result = y-- < 1; // true, aunque y se decrementa después, el valor actual cumple la condición

m) c > d || c > 0

result = c > d || c > 0; // true, 'c' es mayor que 'd' y también tiene un valor numérico mayor que 0

n) 5 \* (i + j) > 'c'

result = 5 \* (i + j) > 'c'; // true, 5 \* (8 + 5) = 65 > 99 ('c' en ASCII)

o) 2 \* x + y == 0

result = 2 \* x + y == 0; // true, 2 \* 0.005 + (-0.01) = 0

p) 2 \* x + (y == 0)

result = 2 \* x + (y == 0); // true, 2 \* 0.005 + (true) = 0.01

q) x + y >= 0

result = x + y >= 0; // true, -0.005 >= 0

r) x < ++y

result = x < ++y; // true, después del incremento, y es positivo

s) -(i + j) != -i + j

result = -(i + j) != -i + j; // false, -13 no es diferente de -13

t) i <= j && i >= c

result = i <= j && i >= c; // true, ambas condiciones son ciertas

u) i > 0 && j < 5

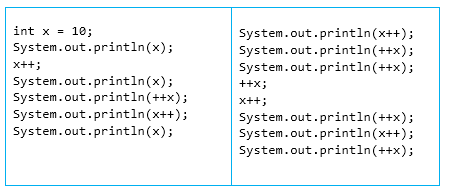
result = i > 0 && j < 5; // true, ambas condiciones son ciertas

v) i > 0 || j < 5

result = i > 0 || j < 5; // true, al menos una condición es cierta

12. Ejercicio con los operadores incremento y decremento: ++ y --.

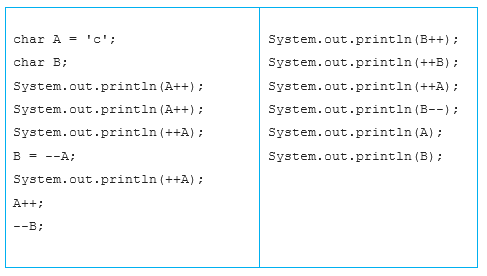
**Ejercicio 1:** Si x es una variable de tipo int con valor 10, determina qué se muestra por pantalla cuando se ejecutan las siguientes instrucciones:



Valor de la X | Salida por pantalla

|  |  |  |  |
| --- | --- | --- | --- |
| int x = 10; | 10 |  |  |
| System.out.println(x); |  | **10** |  |
| x++; | 11 |  |  |
| System.out.println(x); |  | **11** |  |
| System.out.println(++x); | 12 | **12** | Primero se incrementa el valor de x y después se muestra |
| System.out.println(x++); | 13 | **12** | Primero muestra el valor de x y después se realiza el incremento |
| System.out.println(x); |  | **13** |  |
| System.out.println(x++); | 14 | **13** | Primero muestra el valor de x y después se realiza el incremento |
| System.out.println(++x); | 15 | **15** | Primero se incrementa el valor de x y después se muestra |
| System.out.println(++x); | 16 | **16** | Primero se incrementa el valor de x y después se muestra |
| ++x; | 17 |  |  |
| x++; | 18 |  |  |
| System.out.println(++x); | 19 | **19** | Primero se incrementa el valor de x y después se muestra |
| System.out.println(x++); | 20 | **19** | Primero muestra el valor de x y después se realiza el incremento |
| System.out.println(++x); | 21 | **21** | Primero se incrementa el valor de x y después se muestra |

**Ejercicio 2:**Dadas las variables A y B de tipo char,calcula qué se muestra por pantalla cuando se ejecutan las siguientes instrucciones**:**

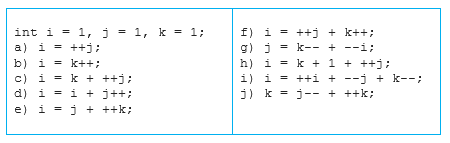


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **Salida por pantalla** |  |
| char A = 'c'; | 'c' |  |  |  |
| char B; |  |  |  |  |
| System.out.println(A++); | 'd' |  | **c** | Primero muestra el valor de A y después se realiza el incremento |
| System.out.println(A++); | 'e' |  | **d** | Primero muestra el valor de A y después se realiza el incremento |
| System.out.println(++A); | 'f' |  | **f** | Primero se incrementa el valor de A y después se muestra |
| B = --A; | 'e' | 'e' |  | Primero se decrementa el valor de A y después se asigna a B |
| System.out.println(++A); | 'f' |  | **f** | Primero se incrementa el valor de A y después se muestra |
| A++; | 'g' |  |  |  |
| --B; |  | 'd' |  |  |
| System.out.println(B++); |  | 'e' | **d** | Primero muestra el valor de B y después se realiza el incremento |
| System.out.println(++B); |  | 'f' | **f** | Primero se incrementa el valor de B y después se muestra |
| System.out.println(++A); | 'h' |  | **h** | Primero se incrementa el valor de A y después se muestra |
| System.out.println(B--); |  | 'e' | **f** | Primero muestra el valor de B y después se realiza el decremento |
| System.out.println(A); |  |  | **h** |  |
| System.out.println(B); |  |  | **e** |  |

**Ejercicio 3:**Un programa Java contiene las siguientes declaraciones y asignaciones iniciales:

**int i = 1, j = 1, k = 1;**

Determina el valor final de las tres variables en cada una de las siguientes instrucciones. Las instrucciones son independientes unas de otras, es decir, el valor inicial de las variables en cada instrucción es i = 1, j = 1, k = 1.



i=1

j=1

k=1

a) i = ++j;

2

2

1

b) i = k++;

1

1

2

c) i = k + ++j;

3

2

1

d) i = i + j++;

2

2

1

e) i = j + ++k;

3

1

2

f) i = ++j + k++;

3

2

2

g) j = k-- + --i;

0

1

0

h) i = k + 1 + ++j;

4

2

1

i) i = ++i + --j + k--;

3

0

0

j) k = j-- + ++k;

1

0

3

13. Ejercicios para practicar la salida de datos por pantalla: System.out.print y System.out.println– I

Relación Nº 1: Ejercicios 1 y 2  
  
Empezamos con una serie de ejercicios para aquellos que están comenzando desde cero con el lenguaje Java y tampoco tienen experiencia con ningún otro lenguaje de programación.

En estos ejercicios básicos inicales se realizan las siguientes instrucciones: declarar variables, asignarles un valor, operar con ellas y mostrar resultados por pantalla.

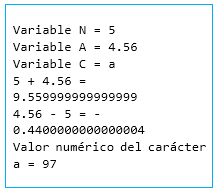
Estos ejercicios son todos de estructura secuencial, es decir, no hay condiciones ni bucles. Tampoco se lee nada por teclado. El objetivo es familiarizarse con la declaración de variables y practicar la salida por consola utilizando los métodos print y println.

**Ejercicio básico inicial 1**:

Escribe un programa Java que realice lo siguiente: declarar una variable N de tipo int, una variable A de tipo double y una variable C de tipo char y asigna a cada una un valor. A continuación muestra por pantalla:

* El valor de cada variable.
* La suma de N + A
* La diferencia de A - N
* El valor numérico correspondiente al carácter que contiene la variable C.

Si por ejemplo le hemos dado a N el valor 5, a A el valor 4.56 y a C el valor ‘a’, se debe mostrar por pantalla:



package bi1;

public class Main {

public static void main(String[] args) {

int N = 5;

double A = 4.56;

char C = 'a';

System.out.println("Variable N = " + N);

System.out.println("Variable A = " + A);

System.out.println("Variable C = " + C);

System.out.println(N + " + " + A + " = " + (N+A));

System.out.println(A + " - " + N + " = " + (A-N));

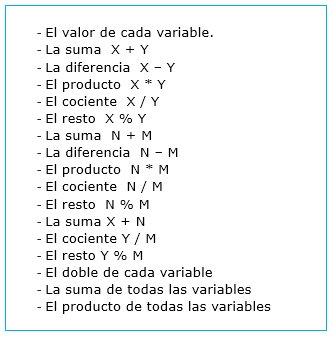
System.out.println("Valor numérico del carácter " + C + " = " + (int)C);

}

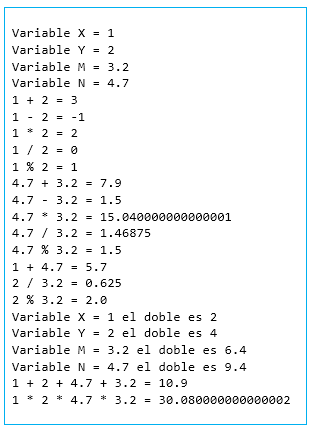
}

**Ejercicio básico inicial 2**:

Escribe un programa Java que realice lo siguiente: declarar dos variables X e Y de tipo int, dos variables N y M de tipo double y asigna a cada una un valor. A continuación muestra por pantalla:



Si por ejemplo le hemos dado a X el valor 1, a Y el valor 2, a M el valor 3.2 y a N el valor 4.7 se debe mostrar por pantalla:



package bi2;

public class Main {

public static void main(String[] args) {

int X = 1, Y = 2;

double M = 3.2, N = 4.7;

System.out.println("Variable X = " + X);

System.out.println("Variable Y = " + Y);

System.out.println("Variable M = " + M);

System.out.println("Variable N = " + N);

System.out.println(X + " + " + Y + " = " + (X+Y));

System.out.println(X + " - " + Y + " = " + (X-Y));

System.out.println(X + " \* " + Y + " = " + X\*Y);

System.out.println(X + " / " + Y + " = " + X/Y);

System.out.println(X + " % " + Y + " = " + X%Y);

System.out.println(N + " + " + M + " = " + (N+M));

System.out.println(N + " - " + M + " = " + (N-M));

System.out.println(N + " \* " + M + " = " + N\*M);

System.out.println(N + " / " + M + " = " + N/M);

System.out.println(N + " % " + M + " = " + N%M);

System.out.println(X + " + " + N + " = " + (X+N));

System.out.println(Y + " / " + M + " = " + Y/M);

System.out.println(Y + " % " + M + " = " + Y%M);

System.out.println("Variable X = " + X + " el doble es " + 2\*X);

System.out.println("Variable Y = " + Y + " el doble es " + 2\*Y);

System.out.println("Variable M = " + M + " el doble es " + 2\*M);

System.out.println("Variable N = " + N + " el doble es " + 2\*N);

System.out.println(X + " + " + Y + " + " + N + " + " + M + " = " + (X+Y+M+N));

System.out.println(X + " \* " + Y + " \* " + N + " \* " + M + " = " + (X\*Y\*M\*N));

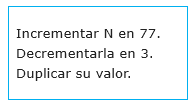
}

}

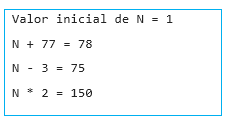
14. Ejercicios para practicar la salida de datos por pantalla: System.out.print y System.out.println– II

**Ejercicio básico inicial 3**

Escribe un programa Java que declare una variable entera N y asígnale un valor. A continuación escribe las instrucciones que realicen los siguientes:



Si por ejemplo N = 1 la salida del programa será:



package bi3;

public class Main {

public static void main(String[] args) {

int N = 1;

System.out.println("Valor inicial de N = " + N);

N+=77;

System.out.println("N + 77 = " + N);

N-=3;

System.out.println("N - 3 = " + N);

N\*=2;

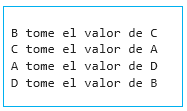
System.out.println("N \* 2 = " + N);

}

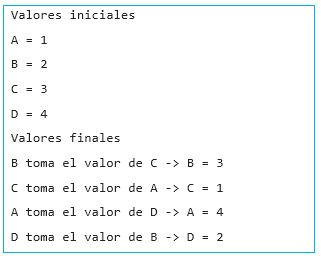
}

**Ejercicio básico inicial 4**

Programa java que declare cuatro variables enteras A, B, C y D y asígnale un valor a cada una. A continuación realiza las instrucciones necesarias para que:



Si por ejemplo A = 1, B = 2, C = 3 y D = 4 el programa debe mostrar:



package bi6;

public class Main {

public static void main(String[] args) {

int A = 1, B = 2, C = 3, D = 4, AUX;

System.out.println("Valores iniciales");

System.out.println("A = " + A);

System.out.println("B = " + B);

System.out.println("C = " + C);

System.out.println("D = " + D);

AUX = B;

B = C;

C = A;

A = D;

D = AUX;

System.out.println("Valores finales");

System.out.println("B toma el valor de C -> B = " + B);

System.out.println("C toma el valor de A -> C = " + C);

System.out.println("A toma el valor de D -> A = " + A);

System.out.println("D toma el valor de B -> D = " + D);

}

}

15. Ejercicios con el operador condicional ( ? : ) - I

En esta entrada vamos a ver tres ejemplos de utilización del operador condicional ? :  
Se trata de usar el operador condicional en lugar de la instrucción condicional if para mostrar por pantalla un mensaje u otro dependiendo de una condición.  
  
**Ejercicio básico inicial 5**

Escribe un programa java que declare una variable A de tipo entero y asígnale un valor. A continuación muestra un mensaje indicando si A es par o impar. Utiliza el operador condicional ( ? : ) dentro del println para resolverlo.

Si por ejemplo A = 14 la salida será



Si fuese por ejemplo A = 15 la salida será:



package bi5;

public class Main {

public static void main(String[] args) {

int A = 15;

System.out.println(A + (A%2==0 ? " es par " : " es impar "));

}

}

**Ejercicio básico inicial 6**

Escribe un programa java que declare una variable B de tipo entero y asígnale un valor. A continuación muestra un mensaje indicando si el valor de B es positivo o negativo. Consideraremos el 0 como positivo. Utiliza el operador condicional ( ? : ) dentro del println para resolverlo.

Si por ejemplo B = 1 la salida será



Si fuese por ejemplo B = -1 la salida será:



package bi6;

public class Main {

public static void main(String[] args) {

int B = -1;

System.out.println(B + (B >= 0 ? " es positivo " : " es negativo "));

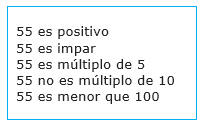
}

}

**Ejercicio básico inicial 7**

Escribe un programa java que declare una variable C de tipo entero y asígnale un valor. A continuación muestra un mensaje indicando si el valor de C es positivo o negativo, si es par o impar, si es múltiplo de 5, si es múltiplo de 10 y si es mayor o menor que 100. Consideraremos el 0 como positivo. Utiliza el operador condicional ( ? : ) dentro del println para resolverlo.

Si por ejemplo C = 55 la salida será



package bi7;

public class Main {

public static void main(String[] args) {

int C = 55;

System.out.println(C + (C >= 0 ? " es positivo " : " es negativo "));

System.out.println(C + (C%2== 0 ? " es par " : " es impar "));

System.out.println(C + (C%5== 0 ? " es múltiplo de 5 " : " no es múltiplo de 5 "));

System.out.println(C + (C%10== 0 ? " es múltiplo de 10 " : " no es múltiplo de 10 "));

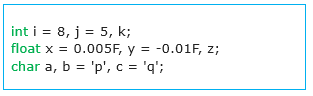
System.out.println(C + (C>100 ? " es mayor que 100 " : " es menor que 100 "));

}

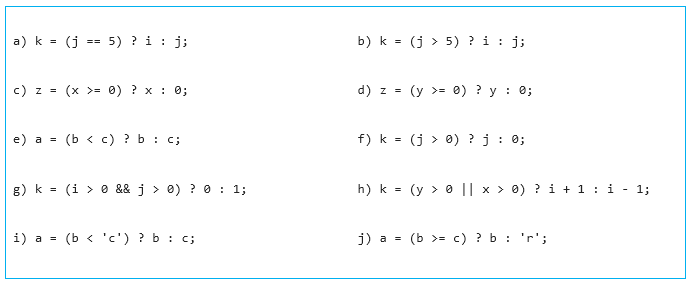
}

16. Ejercicios con el operador condicional ( ? : ) – II

**Ejercicio 1:**Un programa contiene las siguientes declaraciones y asignaciones iniciales de variables:



Determina el valor de cada una de las siguientes expresiones:



a) k = 8

b) k = 5

c) z = 0.005

d) z = 0

e) a = 'p'

f) k = 5

g) k = 0

h) k = 9

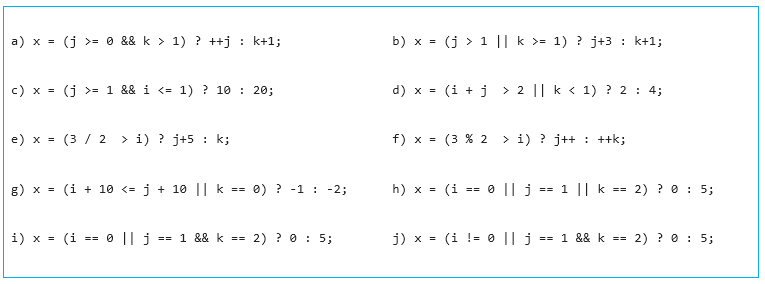
i) a = 'q'

j) a = 'r'

**Ejercicio 2:**Un programa contiene las siguientes declaraciones y asignaciones iniciales de variables:



Determina el valor que toma la variable x al ejecutarse cada una de las siguientes expresiones. Las instrucciones son independientes unas de otras, es decir, el valor inicial para las variables i, j, k en cada instrucción es i = 1, j = 1, k = 1.



a) x = 2

b) x = 4

c) x = 10

d) x = 4

e) x = 1

f) x = 2

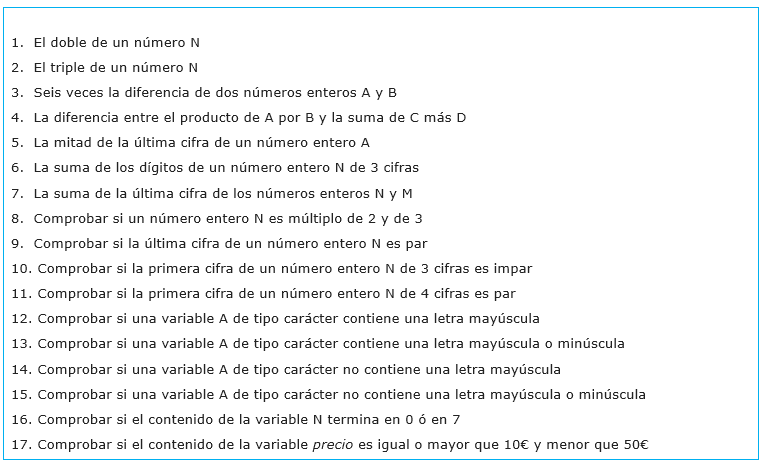
g) x = -1

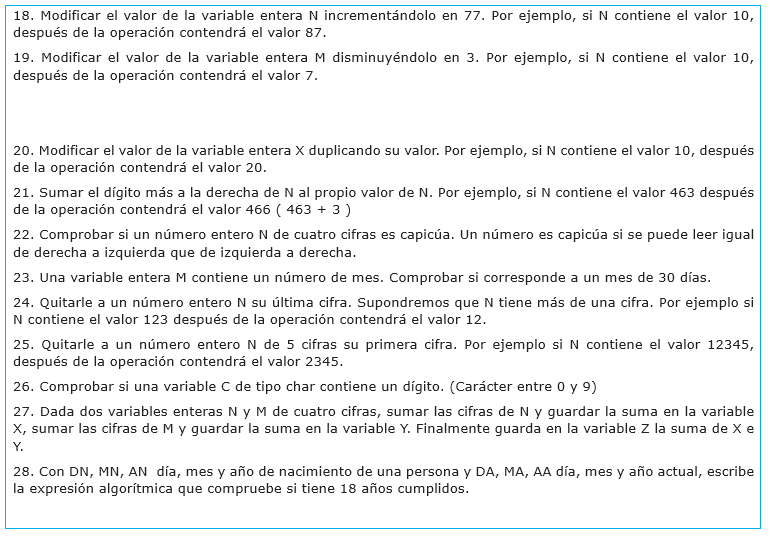
h) x = 0

i) x = 5

j) x = 0

17. Ejercicios sobre expresiones algorítmicas





1.  El doble de un número N

2 \* N

2.  El triple de un número N

3 \* N

3.  Seis veces la diferencia de dos números enteros A y B

6 \* (A – B)

4.  La diferencia entre el producto de A por B y la suma de C más D

A \* B – ( C + D )

5.  La mitad de la última cifra de un número entero A

A % 10 / 2

6.  La suma de los dígitos de un número entero N de 3 cifras

N/100 + N%10 + N/10%10

7.  La suma de la última cifra de los números enteros N y M

N%10 + M%10

8.  Comprobar si un número entero N es múltiplo de 2 y de 3

N%2 == 0 && N%3 == 0

9.  Comprobar si la última cifra de un número entero N es par

N%10%2 == 0

10. Comprobar si la primera cifra de un número entero N de 3 cifras es impar

N/100%2 != 0   También es válida esta solución ->  N/100%2 == 1

11. Comprobar si la primera cifra de un número entero N de 4 cifras es par

N/1000%2 == 0

12. Comprobar si una variable A de tipo carácter contiene una letra mayúscula

A >= ’A’ && A <= ’Z’

13. Comprobar si una variable A de tipo carácter contiene una letra mayúscula o minúscula

A >= ’A’ && A <= ’Z’ || A >= ’a’ && A <= ’z’

14. Comprobar si una variable A de tipo carácter no contiene una letra mayúscula

A < ’A’ || A > ’Z’

15. Comprobar si una variable A de tipo carácter no contiene una letra mayúscula o minúscula

(A < ’A’ || A > ’Z’) && (A < ’a’ || A > ’z’)

También es válida esta solución:

!(A >= ’A’ && A <= ’Z’ || A >= ’a’ && A <= ’z’)

16. Comprobar si el contenido de la variable N termina en 0 ó en 7

N%10 == 0 || N%10 == 7

17. Comprobar si el contenido de la variable *precio* es igual o mayor que 10€ y menor que 50€

precio >= 10 && precio < 50

18. Modificar el valor de la variable entera N incrementándolo en 77.

Por ejemplo, si N contiene el valor 10, después de la operación contendrá el valor 87.

N = N +77   ó   N+=77

19. Modificar el valor de la variable entera M disminuyéndolo en 3.

Por ejemplo, si N contiene el valor 10, después de la operación contendrá el valor 7.

N = N – 3   ó   N-=3

20. Modificar el valor de la variable entera X duplicando su valor.

Por ejemplo, si N contiene el valor 10, después de la operación contendrá el valor 20.

N = N \* 2   ó   N\*=2

21. Sumar el dígito más a la derecha de N al propio valor de N.

Por ejemplo, si N contiene el valor 463 después de la operación contendrá el valor 466 ( 463 + 3 )

N = N + N%10   ó   N+=N%10

22. Comprobar si un número entero N de cuatro cifras es capicúa.

Un número es capicúa si se puede leer igual de derecha a izquierda que de izquierda a derecha.

N/1000 == N%10 && N/100%10 == N/10%10

23. Una variable entera M contiene un número de mes. Comprobar si corresponde a un mes de 30 días.

M == 4 || M == 6 || M == 9 || M == 11

24. Quitarle a un número entero N su última cifra. Supondremos que N tiene más de una cifra.

Por ejemplo si N contiene el valor 123 después de la operación contendrá el valor 12.

N = N/10

25. Quitarle a un número entero N de 5 cifras su primera cifra.

Por ejemplo si N contiene el valor 12345, después de la operación contendrá el valor 2345.

N = N % 10000

26. Comprobar si una variable C de tipo char contiene un dígito. (Carácter entre 0 y 9)

C >= ‘0’ && C <= ‘9’

27. Dada dos variables enteras N y M de cuatro cifras, sumar las cifras de N y guardar la suma en la variable X, sumar las cifras de M y guardar la suma en la variable Y. Finalmente guarda en la variable Z la suma de X e Y.

X = N/1000 + N/100%10 + N/10%10 + N%10

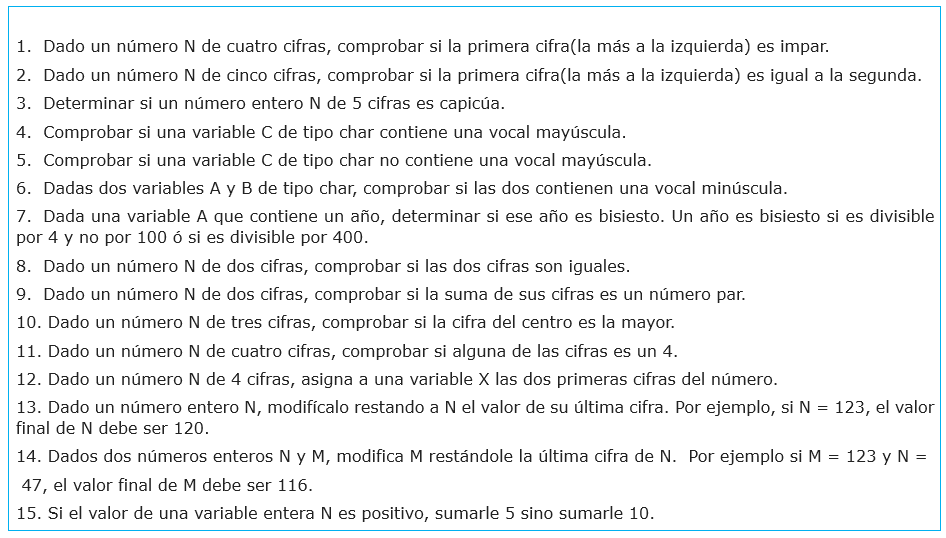
Y = M/1000 + M/100%10 + M/10%10 + M%10

Z = X + Y

28. A partir de las variables enteras DN, MN y AN que contienen el día, mes y año de nacimiento de una persona y las variables DA, MA y AA que contienen el día, mes y año actual, escribe la expresión algorítmica que compruebe si esa persona ya ha cumplido los 18 años.

AA – AN > 18 || AA – AN == 18 && MA > MN || AA – AN == 18 && MA == MN && DA >= DN

18. Ejercicios sobre expresiones algorítmicas – II

Escribe las expresiones algorítmicas equivalentes para cada uno de los siguientes enunciados:

1.  Dado un número N de cuatro cifras, comprobar si la primera cifra(la más a la izquierda) es impar.

N / 1000 % 2 == 0

2.  Dado un número N de cinco cifras, comprobar si la primera cifra(la más a la izquierda) es igual a la segunda

N / 10000 == N / 1000 % 10

3.  Determinar si un número entero N de 5 cifras es capicúa

N / 10000 == N % 10 && N / 1000 % 10 == N / 10 % 10

4.  Comprobar si una variable C de tipo char contiene una vocal mayúscula

C == ‘A’ || C == ‘E’ || C == ‘I’ || C == ‘O’ || C == ‘U’

5.  Comprobar si una variable C de tipo char no contiene una vocal mayúscula

Dos soluciones posibles:

!(C == ‘A’ || C == ‘E’ || C == ‘I’ || C == ‘O’ || C == ‘U’)

C != ‘A’ && C != ‘E’ && C != ‘I’ && C != ‘O’ && C != ‘U’

6.  Dadas dos variables A y B de tipo char, comprobar si las dos contienen una vocal minúscula

(A == ‘a’ || A == ‘e’ || A == ‘i’ || A == ‘o’ || A == ‘u’) &&

B == ‘a’ || B == ‘e’ || B == ‘i’ || B == ‘o’ || B == ‘u’)

7.  Dada una variable A que contiene un año, determinar si ese año es bisiesto. Un año es bisiesto si es divisible por 4 y no por 100 ó si es divisible por 400

A % 4 == 0 && A % 100 != 0 || A % 400 == 0

8.  Dado un número N de dos cifras, comprobar si las dos cifras son iguales

N / 10 == N % 10

9.  Dado un número N de dos cifras, comprobar si la suma de sus cifras es un número par

(N / 10 + N % 10) % 2 == 0

10. Dado un número N de tres cifras, comprobar si la cifra del centro es la mayor

N / 10 % 10 > N / 100 &&  N / 10 % 10 > N % 10

11. Dado un número N de cuatro cifras, comprobar si alguna de las cifras es un 4

N / 1000 == 4 || N / 100 % 10 == 4 || N / 10 % 10 == 4 || N % 10 == 4

12. Dado un número N de 4 cifras, asigna a una variable X las dos primeras cifras del número

X = N / 100

13. Dado un número entero N, modifícalo restando a N el valor de su última cifra. Por ejemplo, si N = 123, el valor final de N debe ser 120

N = N – N % 10 o también N -= N % 10

14. Dados dos números enteros N y M, modifica M restándole la última cifra de N.  Por ejemplo si M = 123 y N = 47, el valor final de M debe ser 116

M = M - N % 10 o también M -= N % 10

15. Si el valor de una variable entera N es positivo, sumarle 5 sino sumarle 10

N = N + (N > 5 ? 5 : 100);   o también N += N > 5 ? 5 : 100;