**Trabajo práctico nro. 1**

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

**Integrantes de la Cátedra:**

* **DOCENTES:**

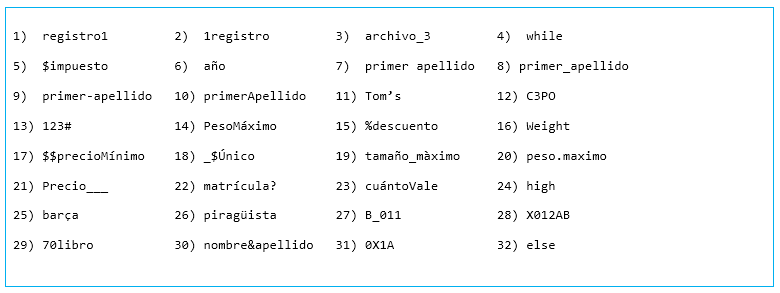
|  |  |  |
| --- | --- | --- |
| **Nombre del Profesor** | **Periodo** | **Cantidad horas materia** |
|  |  | 6 horas |

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: Es valido.

2. 1registro: No es valido ya que comienza con un digito.

3. archivo\_3: Es valido

4. while: No es valido por ser una palabra reservada del lenguaje

5. $impuesto: Es valido

6. año: Es valido

7. primer apellido: Es valido

8. primer\_apellido: Es valido

9. primer-apellido: Es valido

10. primerApellido: Es valido

11. Tom’s: No es valido por que contiene el carácter ‘

12. C3PO : Es valido

13. 123#: No es valido por que contiene el carácter #

14. PesoMáximo: Es valido

15. %descuento: No es valido por que contiene el carácter %

16. Weight: Es valido

17. $$precioMínimo: Es valido

18. \_$Único: Es valido

19. tamaño\_máximo: Es valido

20. peso.maximo: No es valido por que contiene el carácter (.)

21. matrícula?: No es valido por que contiene el carácter ¿

22. cuántoVale: Es valido

23. high: Es valido

24. Precio\_\_: Es valido

25. barҫa: Es valido

26. piragüista: Es valido

27. B\_011: Es valido

28. X012AB: Es valido

29. 70libro: No es valido ya que comienza con un digito.

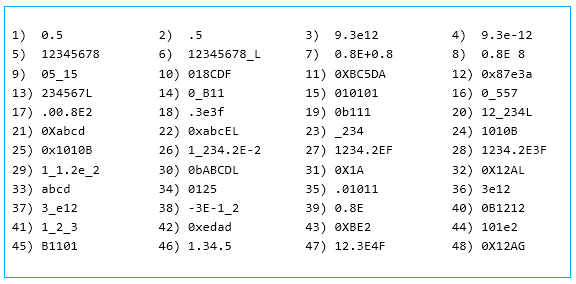
30. nombre&apellido: No es valido por que contiene el carácter &

31. 0X1A: No es valido por que comienza con un digito

32. else: No es valido por que es una palabra reservada

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.



1) 0.5: Es valido de tipo float, esta escrito en el sistema decimal.

2) .5) Es valido de tipo float, esta escrito en el sistema decimal.

3) 9.3e12: Es valido de tipo float, esta escrito en el sistema decimal.

4) 9.3e-12:Es valido de tipo float, esta escrito en el sistema decimal.

5) 12345678: Es valido de tipo int, esta escrito en el sistema decimal.

6) 12345678\_L: No es valido por que aparece el \_ antes del sufijo L.

7) 0.8E + 0.8: No es valido por que la separación de la E y el exponente es un + y dos espacios

8) 0.8E 8:No es valido por que existe separación entre la E y el exponente.

9) 05\_15: Es valido de tipo short, esta escrito en el sistema octal.

10) 018CDF: No es valido por que falta la x después del 0.

11) 0XBC5DA: Es valido de tipo int, esta escrito en el sistema hexadecimal.

12) 0x87e3a: Es valido de tipo int, esta escrito en el sistema hexadecimal.

13) 234567L: Es valido de tipo long, esta escrito en el sistema decimal.

14) 0\_B11: No es valido por que el espacio \_ se encuentra entre el 0 y la B.

15) 010101: Es valido de tipo short, esta escrito en el sistema octal.

16) 0\_557: Es valido de tipo short, esta escrito en el sistema octal.

17) .00.8E2: No es valido por que hay dos “.” en el numero.

18) .3e3f: Es valido de tipo float, esta escrito en el sistema hexadecimal.

19) 0b111: Es valido de tipo de tipo byte, esta escrito en el sistema binario.

20) 12\_234L: Es valido de tipo long, esta escrito en el sistema decimal.

21) 0Xabcd: Es valido de tipo int, esta escrito en el sistema hexadecimal.

22) 0xabcEL: Es valido de tipo long, esta escrito en el sistema hexadecimal.

23) \_234: No es valido por que el \_ esta al principio del numero.

24) 1010B: No es valido por que no contiene el prefijo 0x.

25) 0x1010B: Es valido de tipo int, esta escrito en el sistema hexadecimal.

26) 1\_234.2E-2: Es valido de tipo float, esta escrito en el sistema decimal.

27) 1234.2EF: Es valido de tipo float, esta escrito en el sistema hexadecimal.

28) 1234.2E3F: Es valido de tipo doublé, esta escrito en el sistema hexadecimal.

29) 1\_1.2e\_2: Es valido de short, esta escrito en el sistema decimal.

30) 0bABCDL: No es valido por que tiene el prefijo de binario y esta escrito en hexadecimal

31) 0X1A: Es valido de tipo byte, esta escrito en el sistema hexadecimal.

32) 0X12AL: Es valido de tipo long, esta escrito en el sistema hexadecimal.

33) abcd: No es valido por que no tiene el prefijo 0x.

34) 0125: Es valido de tipo byte, esta escrito en el sistema octal.

35) .01011: Es valido de tipo byte, esta escrito en el sistema decimal.

36) 3e12: Es valido de tipo int, esta escrito en el sistema decimal.

37) 3\_e12: Es valido de tipo int, esta escrito en el sistema decimal.

38) -3E-1\_2: Es valido de tipo int, esta escrito en el sistema decimal.

39) 0.8E: Es valido de tipo float, esta escrito en el sistema hexadecimal.

40) 0B1212: No es valido por que no debería tener el prefijo 0b.

41) 1\_2\_3: Es valido de tipo byte, esta escrito en el sistema decimal.

42) 0xedad: Es valido de tipo int, esta escrito en el sistema hexadecimal

43) 0XBE2: Es valido de tipo short, esta escrito en el sistema hexadecimal.

44) 101e2: Es valido de tipo int, esta escrito en el sistema decimal.

45) B1101: No es valido ya que debería tener el prefijo 0x.

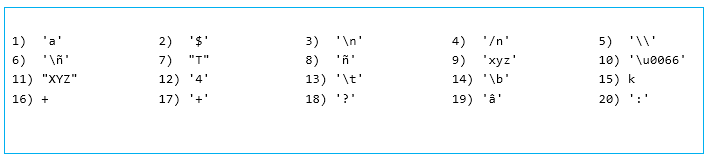
46) 1.34.5: No es valido al tener dos puntos en el mismo numero.

47) 12.3E4F: No es valido ya que debería tener el prefijo 0x.

48) 0X12AG: No es valido ya que la letra g no es parte del sistema hexadecimal.

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.



1) ‘a’: Es valido.

2) ‘$’: Es valido.

3) ‘\n’: Es valido.

4) ‘/n’: No es valido ya que no tiene barra invertida ademas de tener mas de un carácter.

5) ‘\\’:No es valido ya que tiene mas de un carácter.

6) ‘\ñ`’: No es valido ya que tiene mas de un carácter.

7) “T”: No es valido ya que esta entre comillas dobles.

8) `ñ`: Es valido.

9) ‘xyz’: No es valido ya que tiene mas de un carácter.

10) ‘\u0066’: Es valido.

11) ‘XYZ’: No es valido ya que tiene mas de un carácter.

12) ‘4’: Es valido.

13) ‘\t’: No es valido ya que debería haber una n en vez de t.

14) ‘\b’: No es valido ya que debería haber una n en vez de b.

15) k: No es valido ya que no tiene comillas simples.

16) +: No es valido ya que no tiene comillas simples.

17) ‘+’: Es valido.

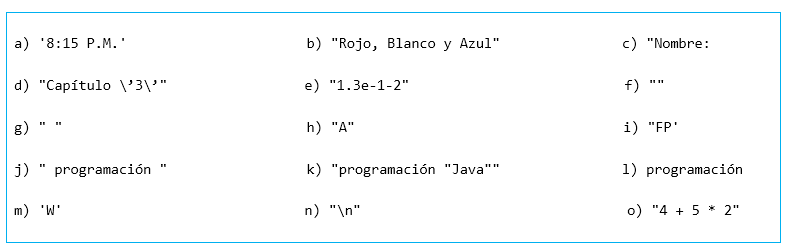
18) ‘?’: Es valido.

19) ‘â’: Es valido

20) ‘z’: Es valido.

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.



1) ‘8:15 P.M.’: No es valido ya que esta entre comillas simples.

2) “Rojo, Blanco y Azul”: Es valido.

3) “Nombre: :No es valido ya que le falta una comilla doble.

4) “Capitulo \’3\’”: Es valido.

5) “1.3e-1-2”: Es valido.

6) “”: Es valido.

7) “ “:Es valido.

8) “A”: Es valido.

9) “FP’: No es valido ya que lo encierra una comilla doble y simple.

10) “programación “Java””: Es valido.

11) programación: No es valido por que no esta entre comillas dobles.

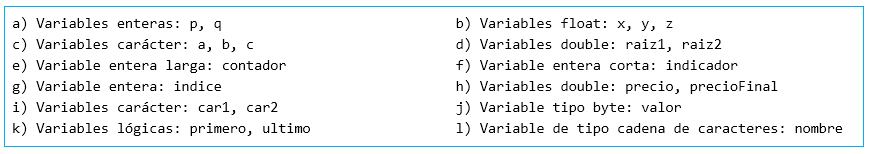
12) ‘W’: No es valido por que esta entre comillas simples.

13) “\n”: Es valido.

14) “4 + 5 \* 2”: Es valido.

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.



1) int p = 70000, int q = 80000

2) char a = ‘a’, char b = ‘b’, char c = ‘c’

3) long contador = 9e20

4) int índice = 90000

5) char car1 = ‘1’, char car2 = ‘2’

6) boolean primero = true, boolean ultimo = false

7) float x = 4.5, float y = 6.7, float z = 9.3

8) double raiz1 = 3.4e40, double raiz2 = -3.4e40

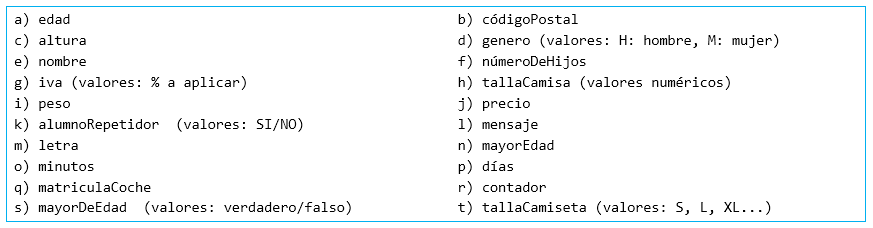
9) short indicador = 2000

10) double precio = 4.5e41, double precioFinal = -4.5e41

11) byte valor = 100

12) string nombre = “Matias”

**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.



1) byte edad.

2) byte altura.

3) string nombre.

4) byte iva.

5) short peso.

6) boolean alumnoRepetidor.

7) char letra.

8) int minutos.

9) int matriculaCoche.

10) boolean mayorDeEdad.

11) int codigoPostal.

12) string genero.

13) byte numeroDeHijos

14) byte tallaCamisa.

15) int precio.

16) string mensaje.

17) boolean mayorEdad.

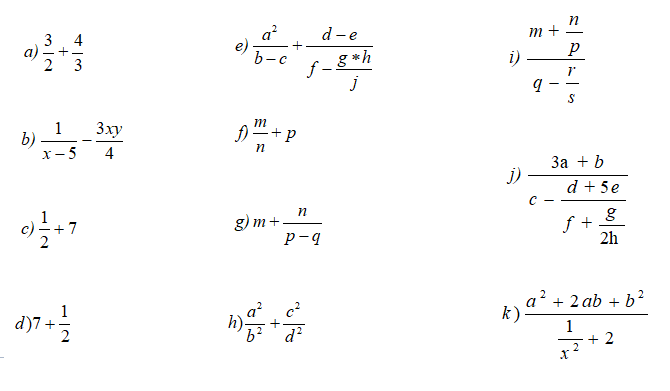
18) int días.

19) int contador.

20) string tallaCamiseta.

6. Convertir expresiones algebráicas en expresiones algorítmicas

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



1) 3 / 2 + 4 / 3

2) 1 / x – 5 – 3 \* x \* y / 4

3) 1 / 2 + 7

4) 7 + 1 / 2

5) Math.pow(a , 2) / b – c + (d – e )/f - g \* h / j

6) m / n + p

7) m + n / (p – q)

8) Math.pow(a , 2) / Math.pow(b , 2) + Math.pow(c , 2) / Math.pow(d , 2)

9) m + n / p+ q – r / s

10) (3 \* a + b) / (c – (d + 5 \* e)) / (f + g / (2 \* h))

11) (Math.pow(a , 2) + 2 \* a \* b + Math.pow(b , 2)) / (1 / Math.pow(x , 2) + 2)

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:



1) b / 2 – 4 \* a \* c == -158.5

2) 3 \* x \* y – 5 \* x +12 \* x – 17 == 111

3) (b + d) / (c + 4) == -10

4) (x \* y) / y + 2 == 10

5) 1 / y + 3 \* x / z + 1 == -3 probar en Java cuando te debería dar un real

6) 1 / (y + 3) + x / y + 1 == 3

7) 1 / y + (3 + x) / (y +1) == 3

**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 b) 2 \* y + 3 \* (x – z)  c) x / y d) x % y  e) x / (y + z) f) (x / y) + z  g) 2 \* x / 3 \* y h) 2 \* x / (3 \* y)  i) x \* y % z j) x \* (y % z)  k) 3 \* x – z – 2 \* x l) 2 \* x / 5 % y  m) x - 100 % y % z n) x - y - z \* 2 |

**1) x + y + z == 86.3**

**2) 2 \* y + 3 \* (x – z) == 286.6**

**3) x / y == 25.1428**

**4) x % y == 3.5**

**5) x / (y + z) == -51.7647**

**6) (x / y) + z == 19,9428**

**7) 2 \* x / 3 \* y == 16,7619**

**8) 2 \* x / (3 \* y) == 16,7619**

**9) x \* y % z == -4.0011**

**10) x \* (y % z) == -149.6176**

**11) 3 \* x – z – 2 \* x == 93.2**

**12) 2 \* x / 5 % y == 0.2**

**13) x – 100 % y % z == 91.2**

**14) x – y – z \* 2 == 94.9**

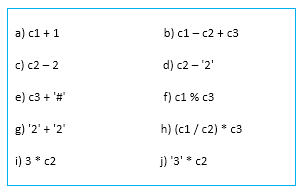
**Ejercicio 3.**Si c1, c2 y c3 son variables de tipo char con valores c1=’E’, c2=’5’, c3=’?’, determina el valor numérico de las siguientes expresiones aritméticas. Para resolverlo necesitas saber el valor numérico correspondiente a esos caracteres según la tabla ASCII:

’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 == 82

i) 3 \* c2 == 159

j) ‘3’ \* c2 == 2703

9. Determinar el tipo del resultado en operaciones aritméticas

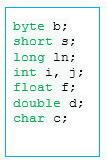
En aquellas expresiones aritméticas en las que intervienen operandos de distinto tipo (int, double, char, etc.) java convierte el tipo de los operandos que intervienen al tipo del operando de mayor precisión y este será el tipo del resultado obtenido en la operación.

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.

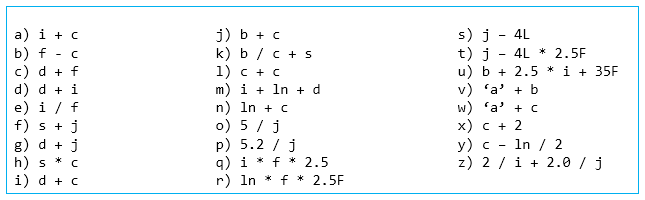
Debemos tener en cuenta que cuando se realiza la conversión temporal de tipos, los operandos de tipo short, byte y char se convierten a int.

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:



1) i + c: int

2) f – c: float

3) d + f: float

4) d + i: int

5) i / f: float

6) s + j: int

7) d + j: int

8) s \* c: int

9) d + c: int

10) b + c: int

11) b / c + s: int

12) c + c: int

13) i + ln + d: long

14) ln + c: long

15) 5 / j: int

16) 5.2 / j: float

17) i \* f \* 2.5: doublé

18) ln \* f \* 2.5F: doublé

19) j – 4L : int

20) j – 4L \* 2.5F: doublé

21) b + 2.5 \* i + 35F: doublé

22) ‘a’ + b: int

23) ‘a’ + c: int

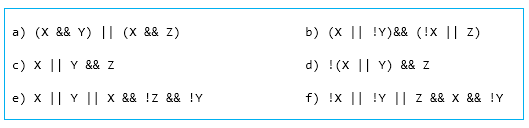
24) c + 2: int

25) c – ln / 2: long

26) 2 / i + 2.0 / j: 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:



1) (x && y) || (x && z) == true

2) (x || ¡y) && (¡x || z) == true

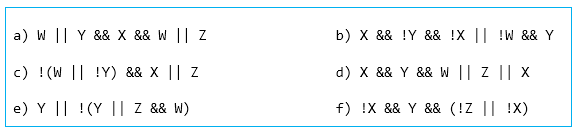
3) x || y && z == true

4) ¡(x || y) && z == false

5) x || y || x && ¡z && ¡y == true

6) ¡x || ¡y || z && x && ¡y == true

**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:



1) w || y && x && w || z == false

2) x && ¡y && ¡x || ¡w && y == true

3) ¡(w || ¡y) && x || z == false

4) x && y && w || z || x == true

5) y || ¡(y || z && w) == true

6) ¡x && y && (¡z || ¡x) == false

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:

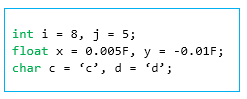
|  |
| --- |
| Precedencia de los operadores lógicos |
|  |

11. Ejercicios sobre operadores relacionales

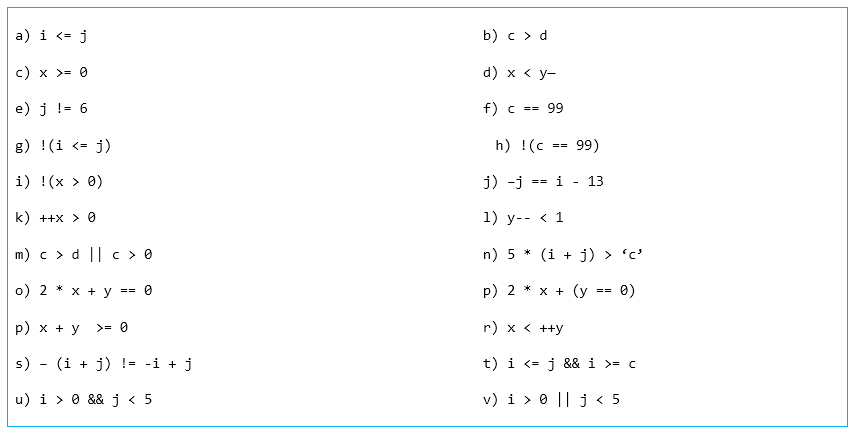
En estos ejercicios practicaremos con los operadores Java relacionales. En algunos de los ejercicios, los operadores relacionales aparecerán en expresiones en los que también intervienen el resto de operadores Java: operadores aritméticos, lógicos y los operadores incremento y decremento.

**Ejercicio:**

A partir de las siguientes declaraciones de variables:



Determina el valor de las siguientes expresiones:



1) i <= j == false

2) c > d == false

3) x >= 0 == true

4) x < -y == true

5) j ¡= 6 == true

6) c == 99 == true

7) ¡(i <= j) == true

8) ¡(c == 99) == false

9) ¡(x > 0) == false

10) -j == i – 13 == true

11) ++x > 0 == true

12) y-- < 1 == true

13) c > d || c > 0 == true

14) 5 \* (i + j) > ‘c’ == false

15) 2 \* x + y == 0 == true

16) 2 \* x + (y == 0) == 0.1F

17) x + y >= 0 == false

18) x < ++ y true

19) – (i + j) ¡= -i + j == true

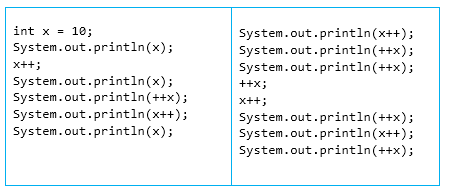
20) i <= j && i >= c == false

21) i > 0 && j < 5 == false

22) i > 0 || j < 5 == true

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:



10

11

12

12

13

13

15

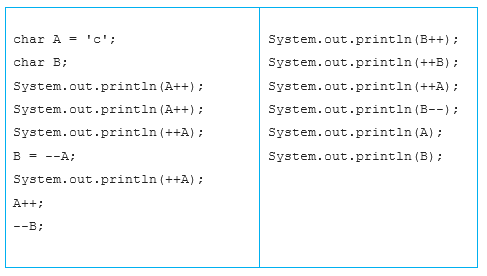
16

19

19

21

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



99

100

102

102

100

102

104

102

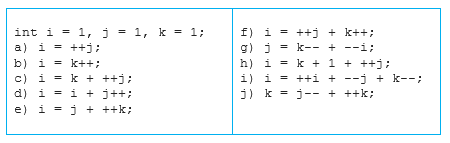
104

101

**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.



1) i = ++j; i == 2, j == 2, k == 1.

2) i = k++, i == 1, j == 1, k == 2.

3) i = k + ++j; i == 3, j == 2, k == 1.

4) i = j + ++k; i == 3, j == 1, k == 2.

5) i = ++j + k++; i i == 3, j == 2, k == 2.

6) j = k-- + --i; i == 0, j == 1, k == 0.

7) i = k + 1 + ++j; i == 3, j == 2, k == 1.

8) i = ++i + --j + k--; i == 3, j == 0, k == 0.

9) k = j-- + ++k; i ==1, j == 0, k ==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.

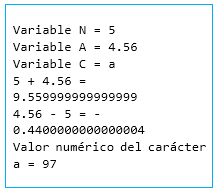
Estos ejercicios son todos de estructura secuencial, es decir, no hay condiciones ni bucles. Tampoco se lee nada por teclado. El objetivo es familiarizarse con la declaración de variables y practicar la salida por consola utilizando los métodos print y println.

**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:



int N = 4;

double A = 3.6E50;

char C = 'h';

System.out.println(N);

System.out.println(A);

System.out.println(C);

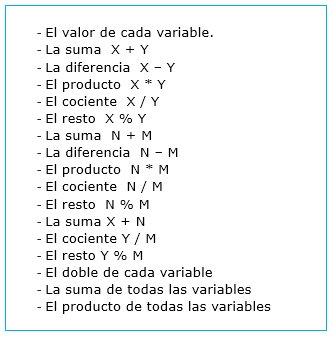
System.out.println("N + A = "+(N + A));

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

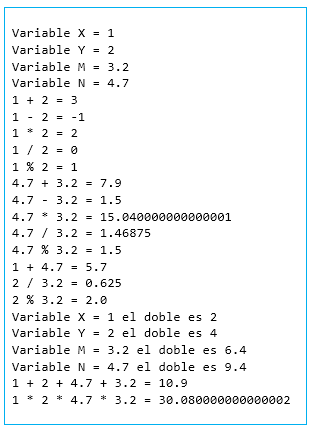
System.out.println("C = "+(C \* 1));

**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:



int x = 4;

int y = 9;

double n = 7.8;

double m = 4.6;

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("2 \* " + x + " = " + (2 \* x));

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

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

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

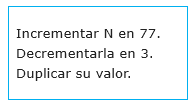
System.out.println(x + " + " + y + " + " + n + " + " + m + " = " + (x + y + n + m));

System.out.println(x + " \* " + y + " \* " + n + " \* " + m + " = " + (x \* y \* n \* m));

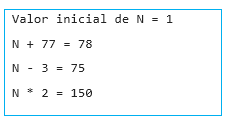
14. Ejercicios para practicar la salida de datos por pantalla: System.out.print y System.out.println – II

**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:



int N = 3;

N = N + 77;

System.out.println(N);

N = N - 3;

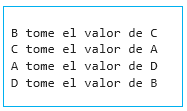
System.out.println(N);

N = N \* 2;

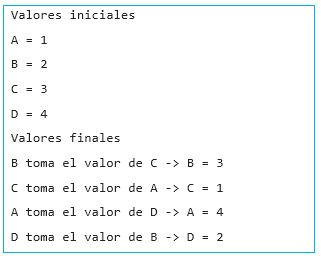
System.out.println(N);

**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:



int A = 3;

int B = 4;

int C = 7;

int D = 8;

B = C;

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

C = A;

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

A = D;

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

D = B;

System.out.println("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:



int A = 6;

String mensaje = (A % 2 == 0) ? "es par" : "es impar";

System.out.println(mensaje);

**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:



int B = 5;

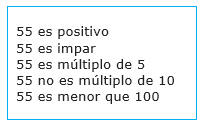
String signo = (B >= 0) ? "Es positivo" : "Es negativo";

System.out.println(signo);

**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



int C = 8;

String mensaje1 = (C >= 0) ? "Es positivo" : "Es negativo";

String mensaje2 = (C % 2 == 0) ? "Es par" : "Es impar";

String mensaje3 = (C % 5 == 0) ? "Es multiplo de 5" : "No es multiplo de 5";

String mensaje4 = (C % 10 == 0) ? "Es multiplo de 10" : "No es multiplo de 10";

String mensaje5 = (C >= 100) ? "Es mayor a 100" : "No es mayor a 100";

System.out.println(mensaje1);

System.out.println(mensaje2);

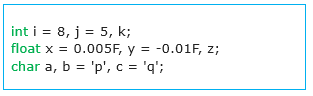
System.out.println(mensaje3);

System.out.println(mensaje4);

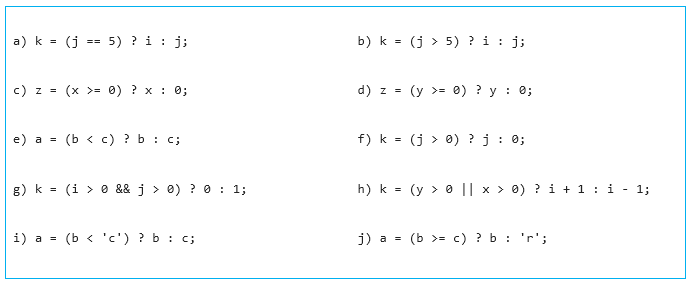
System.out.println(mensaje5);

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:



**1) k = (j == 5) ¿ i : j; k == 8;**

**2) k = (j > 5) ¿ i : j; k == 5;**

**3) z = (x >= 0) ¿ x : 0; z == 0.005F**

**4) z = (y >= 0) ¿ y : 0; z == 0;**

**5) a = (b < c) ¿ b : c; a == ‘p’;**

**6) k = (j > 0) ¿ j : 0; k == 5;**

**7) k = (i > 0 && j > 0) ¿ 0 : 1; k == 0;**

**8) k = (y > 0 || x > 0) ¿ i +1 : i -1; k == 9;**

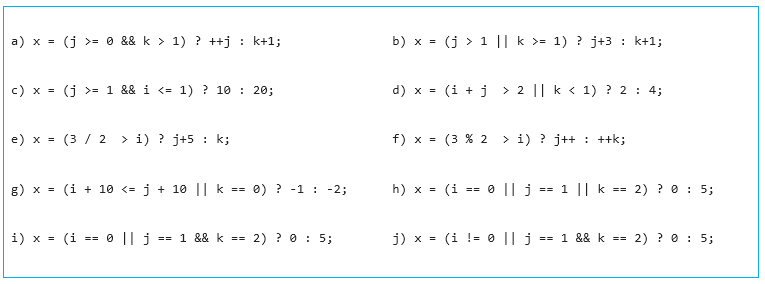
**9) a = (b < ‘c’) ¿ b : c; a == ‘q’;**

**10) a = (b >= c) ¿ b : ‘r’; 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.



1) x = (j >= 0 && k > 1) ¿ ++j : k+1; x == 2;

2) x = (j > 1 || k >= 1) ¿ j+3 : k+1; x == 4;

3) x = (j >= 1 && i <= 1) ¿ 10 : 20; x == 10;

4) x = (i + j > 2 || k < 1) ¿ 2 : 4; x == 4;

5) x = (3 / 2 > i ) ¿ j+5 : k; x == 6;

6) x = (3 % 2 > i) ¿ j++ : ++k; x == 2;

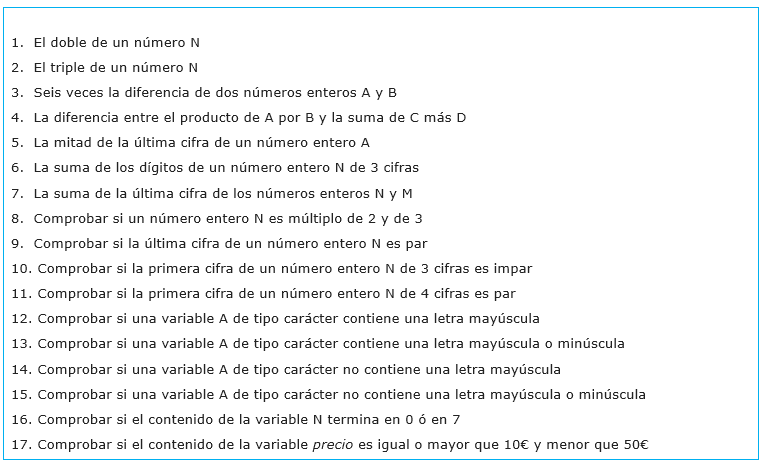
7) x = (i + 10 <= j + 10 || k == 0) ¿ -1 : -2; x == -1;

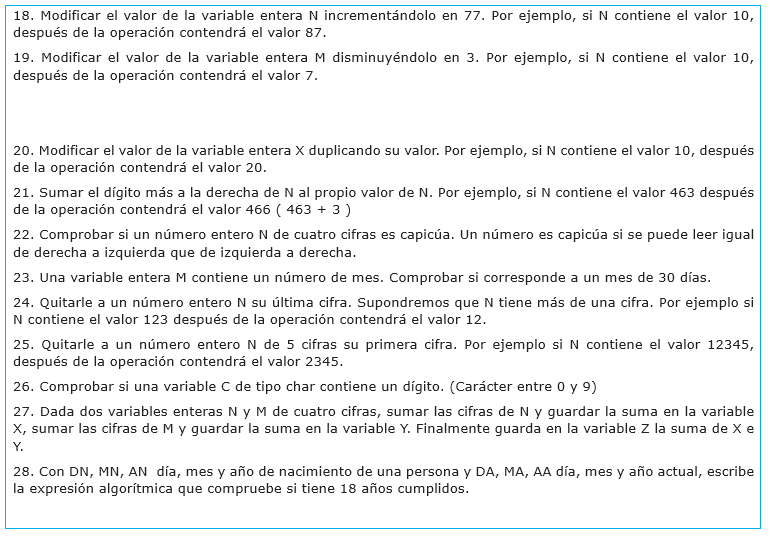
8) x = (i == 0 || j == 1 || k == 2) ¿ 0 : 5; x == 0;

9) x = (i == 0 || j == 1 && k == 2) ¿ 0 : 5; x == 5;

10) x = (i ¡= 0 || j == 1 && k == 2) ¿ 0 : 5; x == 0;

17. Ejercicios sobre expresiones algorítmicas





1)2 \* N

2) 3 \* N

3) 6 \* (A – B)

4) (A \* B) – (C + D)

5) A % 10 / 2

6) N % 10 + N / 100 + (N / 10) % 10

7) N % 10 + M % 10

8) N % 2 == 0 && N % 3 == 0

9) (N % 10) % 2 == 0

10) (N / 100) % 2 ¡= 0

11) (N / 1000) % 2 == 0

12) A >= ‘A’ && A <= ‘Z’

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

14) A < `A` || A > ‘Z’

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

16) N % 10 == 0 || N % 10 == 7

17) precio >= 10 && precio < 50

18) N = N + 77

19) M = M – 3

20) X = X \* 2

21) N = N + (N % 10)

22) N / 1000 == N % 100 && (N / 100) % 10 == (N / 10) % 10

23) M == 4 || M == 6 || M == 9 || M == 11

24) N = N / 10

25) N = N – ((N/ 10000) \* 10000))

26) C >= ‘0’ && C <= ‘9’

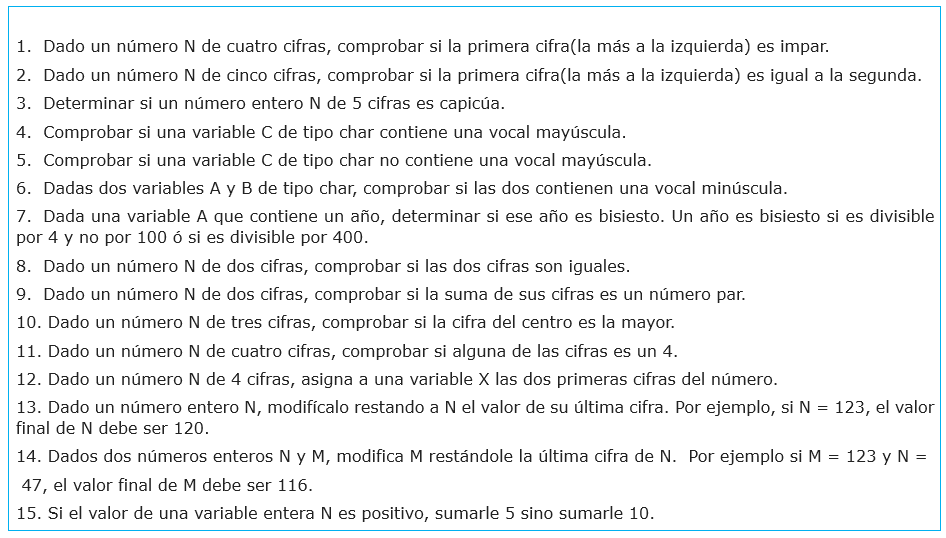
27) X = N % 10 + ((N / 10) % 10) + ((N / 100) % 10) + N / 1000

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

Z = X + Y

28) (AA + MA / 12 + DA / 365) – ( AN + MN / 12 + DN / 365) >= 18

18. Ejercicios sobre expresiones algorítmicas – II

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

1) (N / 1000) % 2 ¡= 0

2) N / 1000 == (N / 100) % 10

3) N / 10000 == N % 10 && (N / 1000) % 10 == (N / 10) % 10

4) C >= ‘A’ && C <= `Z`

5) C < ‘A’ || C > ‘Z’

6) (A >= ‘a’ && A <= ‘z’) && (B >= ‘a’ && B <= ‘z’)

7) (A % 4 == 0 && A % 100 ¡= 0) || A % 400 == 0

8) N % 10 == N / 10

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

10) (N / 10) % 10 > N % 10 && (N / 10) % 10 > N / 100

11) N % 10 == 4 || (N / 10) % 10 == 4 || (N / 100) % 10 == 4 || N / 1000 == 4

12) X = N / 1000 + (N / 100) % 10

13) N = N – ( N % 10)

14) M = M – (N % 10)

15) (N >= 0) ¿ N = N + 5 : N = N + 10