**Ejercicio 1:**

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

1) registro1 2) 1registro 3) archivo\_3 4) while

5) $impuesto 6) año 7) primer apellido 8) primer\_apellido

9) primer-apellido 10) primerApellido 11) Tom’s 12) C3PO

13) 123# 14) PesoMáximo 15) %descuento 16) Weight

17) $$precioMínimo 18) \_$Único 19) tamaño\_màximo 20) peso.maximo

21) Precio\_\_\_ 22) matrícula? 23) cuántoVale 24) high

25) barça 26) piragüista 27) B\_011 28) X012AB

29) 70libro 30) nombre&apellido 31) 0X1A 32) else

1) válido

2) No válido: No puede comenzar con un dígito

3) válido

4) No válido: palabra reservada

5) válido

6) válido

7) No válido: no pueden contener espacios, serian 2 identificadores

8) Válido

9) No válido: no puede contener carácter especial –

10) Válido

11) No válido: no puede contener carácter especial ‘

12) válido

13) No válido: no puede comenzar con un dígito

14) Válido

15) No válido: no puede contener carácter especial %

16) válido

17) válido

18) Válido

19) válido

20) No válido: no puede contener carácter especial .

21) Válido

22) No válido: no puede contener carácter especial ?

23) Válido

24)válido

25) válido

26) válido

27) válido

28) válido

29)No válido: no puede comenzar con un dígito

30)No válido: no puede contener carácter especial &

31) No válido: no puede comenzar con un dígito

32) No válido: palabra reservada

**Ejercicio 2:**

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 por qué no lo es.

1) 0.5 2) .5 3) 9.3e12 4) 9.3e-12

5) 12345678 6) 12345678\_L 7) 0.8E+0.8 8) 0.8E 8

9) 05\_15 10) 018CDF 11) 0XBC5DA 12) 0x87e3a

13) 234567L 14) 0\_B11 15) 010101 16) 0\_557

17) .00.8E2 18) .3e3f 19) 0b111 20) 12\_234L

21) 0Xabcd 22) 0xabcEL 23) \_234 24) 1010B

25) 0x1010B 26) 1\_234.2E-2 27) 1234.2EF 28) 1234.2E3F

29) 1\_1.2e\_2 30) 0bABCDL 31) 0X1A 32) 0X12AL

33) abcd 34) 0125 35) .01011 36) 3e12

37) 3\_e12 38) -3E-1\_2 39) 0.8E 40) 0B1212

41) 1\_2\_3 42) 0xedad 43) 0XBE2 44) 101e2

45) B1101 46) 1.34.5 47) 12.3E4F 48) 0X12AG

1) double decimal

2) double decimal

3) double decimal

4) double decimal

5) int decimal

6) No válido: Posición no válida para \_

7) No válido: el exponente no puede ser decimal

8) No válido: hay un espacio entre el 8 y la E

9) int octal

10) No válido: no es octal porque contiene los caracteres cdf y tampoco es hexadecimal porque no comienza con 0x

11) int hexadecimal

12) int hexadecimal

13) long decimal

14) No es válido: el \_ solo puede estar entre dígitos

15) int octal

16) int octal

17) No es válido: no puede contener más de un punto . que separe la parte decimal

18) float decimal

19) int binario

20) long decimal

21) int hexadecimal

22) long hexadecimal

23) No válido: el \_ solo puede aparecer entre dígitos

24) No válido: para ser binario debería comenzar con 0B

25) int hexadecimal

26) double decimal

27) No válido: falta el exponente luego de E

28) float decimal

29) No válido: el carácter \_ solo puede estar entre dígitos.

30) No válido: para ser binario solo debe contener los dígitos 0 y 1, podría ser long hexadecimal si comenzara con 0x

31) int hexadecimal

32) long hexadecimal

33) No válido: debería comenzar por 0x para ser hexadecimal

34) int octal

35) double decimal

36) double decimal

37) No válido: el \_ solo puede estar entre dígitos

38) double decimal

39) No válido: falta el exponente

40) No válido: los binarios solo contienen 0 y 1

41) int decimal

42) int hexadecimal

43) int hexadecimal

44) double decimal

45) No válido: para ser binario debería comenzar con 0B

46) No válido: no puede contener más de un punto separador del decimal

47) float decimal

48) No válido: G no es un dígito aceptable en números hexadecimales.

**Ejercicio 3:**

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

1) 'a' 2) '$' 3) '\n' 4) '/n' 5) '\\'

6) '\ñ' 7) "T" 8) 'ñ' 9) 'xyz' 10) '\u0066'

11) "XYZ" 12) '4' 13) '\t' 14) '\b' 15) k

16) + 17) '+' 18) '?' 19) 'â' 20) ':'

1) válido

2) válido

3) válido

4) No válido: contiene más de 1 carácter y no es una secuencia de escape

5) Válido

6) No válido: contiene más de 1 carácter y no es una secuencia de escape

7) No válido: debe ir entre comillas simples

8) válido

9) No válido: contiene más de un carácter

10) Válido

11) No válido: es una cadena

12) válido

13) válido

14) válido

15) No válido: debe ir entre comillas simples

16) No válido: debe ir entre comillas simples

17) válido

18) válido

19) válido

20) válido

**Ejercicio 4:**

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

a) '8:15 P.M.' b) "Rojo, Blanco y Azul" c) "Nombre:

d) "Capítulo \’3\’" e) "1.3e-1-2" f) ""

g) " " h) "A" i) "FP'

j) " programación " k) "programación "Java"" l) programación

m) 'W' n) "\n" o) "4 + 5 \* 2"

a) No es válido: debe ir entre comillas dobles

b) válido

c) No válido: falta el cierre de comillas

d) válido

e) válido

f) válido

g) válido

h) válido

i) No válido: debe cerrarse con comillas dobles

j) válido

k) No válido: debe utilizarse la secuencia de escape \”

l) No válido: debe estar entre comillas dobles

m) No válido: debe estar entre comillas dobles

n) válido

o) válido

**Ejercicio 5.1:**

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 b) Variables float: x, y, z

c) Variables carácter: a, b, c d) Variables double: raiz1, raiz2

e) Variable entera larga: contador f) Variable entera corta: indicador

g) Variable entera: indice h) Variables double: precio, precioFinal

i) Variables carácter: car1, car2 j) Variable tipo byte: valor

k) Variables lógicas: primero, ultimo l) Variable de tipo cadena de caracteres: nombre

a) int p = 45, q = 6;

b) float x=12.5f, y=.2f, z=452.3F;

c) char a= ‘b’, b=’b’, c=’r’;

d) double raiz1=40.3, raiz2=.45;

e) long contador=0;

f) short indicador=10;

g) int índice=10;

h) double precio=25, precioFinal=1524.8;

i) char car1=’f’, car2=’\n’

j) byte valor=43

k) boolean primero= false, ultimo=true;

l) String nombre=”pedro”

**Ejercicio 5.2:**

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 b) códigoPostal

c) altura d) genero (valores: H: hombre, M: mujer)

e) nombre f) númeroDeHijos

g) iva (valores: % a aplicar) h) tallaCamisa (valores numéricos)

i) peso j) precio

k) alumnoRepetidor (valores: SI/NO)

l) mensaje

m) letra n) mayorEdad

o) minutos p) días

q) matriculaCoche r) contador

s) mayorDeEdad (valores: verdadero/falso)

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

a) int edad;

b) String códigoPostal;

c) double altura;

d) char genero;

e) String nombre;

f) int númeroDeHijos;

g) double iva;

h) int tallaCamisa;

i) double peso;

j) double precio;

k) String alumnoRepetidor;

l) String mensaje;

m) char letra;

n) int mayorEdad;

o) int minutos;

p) int días;

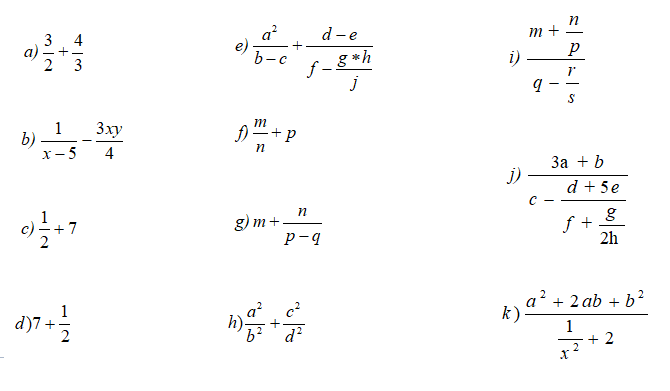
q) String matriculaCoche;

r) int contador;

s) boolean mayorEdad;

t) String tallaCamiseta;

**Ejercicio 6:**

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

a) 3/2+4/3

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

c) ½+7

d)7+1/2

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

f) m/n+p

g) m+n/(p-q)

h) a^2/b^2+c^2/d^2

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

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

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

**Ejercicio 7:**

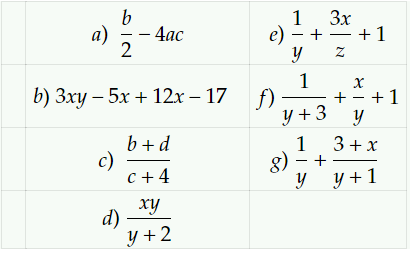
Convierte en expresiones algebráicas las siguientes expresiones algorítmicas.

a) b/2-4\*a\*c e) 1/y+3\*x/z+1

b) 3\*x\*y-5\*x+12\*x-17 f) 1/(y+3)+x/y+1

c) (b+d)/(c+4) g) 1/y+(3+x)/(y+1)

d) (x\*y)/y+2



**Ejercicio 8.1:**

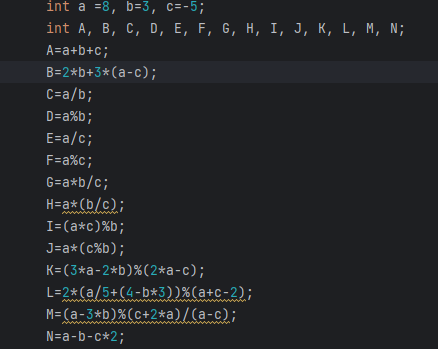
Si a, b y c son variables enteras con valores a=8, b=3, c=-5, determina el valor de las siguientes expresiones aritméticas:

a) a + b + c b) 2 \* b + 3 \* (a – c)

c) a / b d) a % b

e) a / c f) a % c

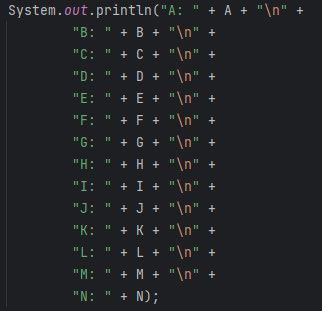
g) a \* b / c h) a \* (b / c)

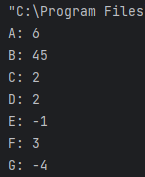


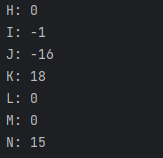
i) (a \* c) % b j) a \* (c % b)

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

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







**Ejercicio 8.2:**

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)

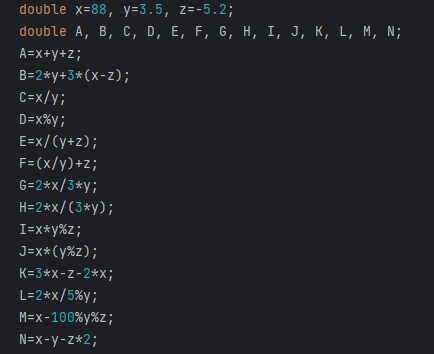
c) x / y d) x % y

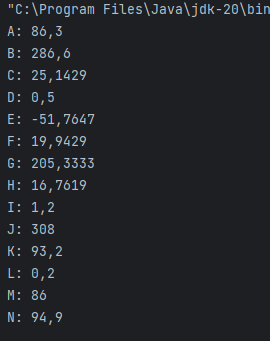
e) x / (y + z) f) (x / y) + z

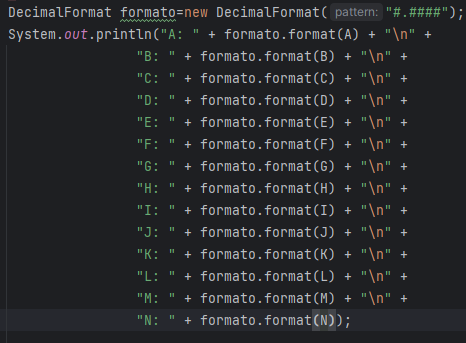
g) 2 \* x / 3 \* y h) 2 \* x / (3 \* y)

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





**Ejercicio 8.3:**

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 b) c1 – c2 + c3

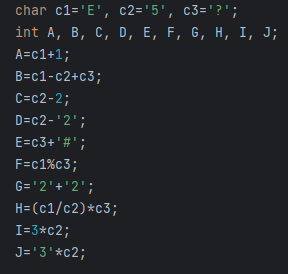
c) c2 – 2 d) c2 – '2'

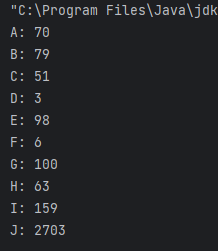
e) c3 + '#' f) c1 % c3

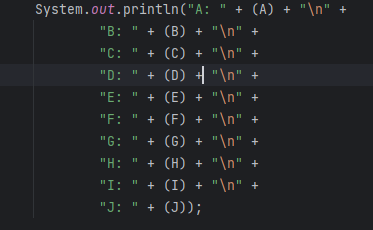
g) '2' + '2' h) (c1 / c2) \* c3

i) 3 \* c2 j) '3' \* c2

‘3’=51; ‘2’=50; ‘#’=35



****

****

**Ejercicio 9:**

A partir de las siguientes declaraciones de variables:

byte b;

short s;

long ln;

int i, j;

float f;

double d;

char c;

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

a) i + c j) b + c s) j – 4L

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

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

d) d + i m) i + ln + d v) ‘a’ + b

e) i / f n) ln + c w) ‘a’ + c

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

g) d + j p) 5.2 / j y) c – ln / 2

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

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

a) int

b) float

c) double

d) double

e) float

f) int

g) double

h) int

i) double

j) int

k) int

l) int

m) double

n) ln

o) double

p) double

q) double

r) float

s) ln

t) float

u) double

v) int

w) int

x) int

y) int

z) double

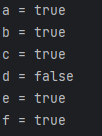
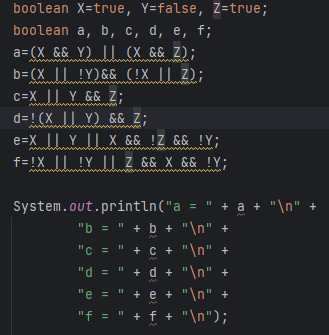
**Ejercicio 10.1:**

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) b) (X || !Y)&& (!X || Z)

c) X || Y && Z d) !(X || Y) && Z

e) X || Y || X && !Z && !Y f ) !X || !Y || Z && X && !Y



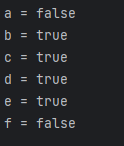
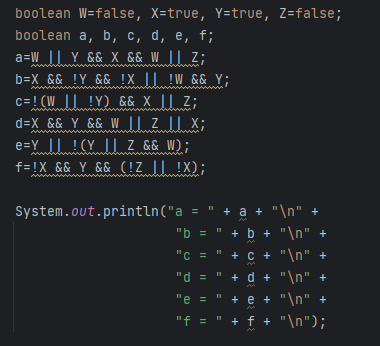
**Ejercicio 10.2:**

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 b) X && !Y && !X || !W && Y

c) !(W || !Y) && X || Z d) X && Y && W || Z || X

e) Y || !(Y || Z && W) f) !X && Y && (!Z || !X)

**Ejercicio 11:**

A partir de las siguientes declaraciones de variables:

int i = 8, j = 5;

float x = 0.005F, y = -0.01F;

char c = ‘c’, d = ‘d’;

Determina el valor de las siguientes expresiones:

a) i <= j b) c > d

c) x >= 0 d) x < y--

e) j != 6 f) c == 99

g) !(i <= j) h) !(c == 99)

i) !(x > 0) j) –j == i - 13

k) ++x > 0 l) y-- < 1

m) c > d || c > 0 n) 5 \* (i + j) > ‘c’

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

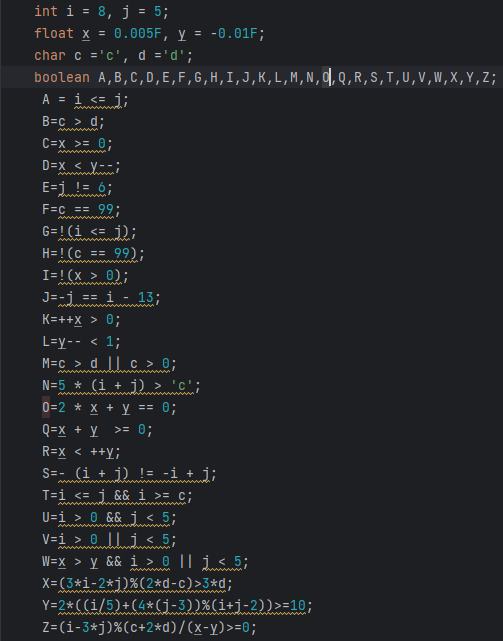
q) x + y >= 0 r) x < ++y

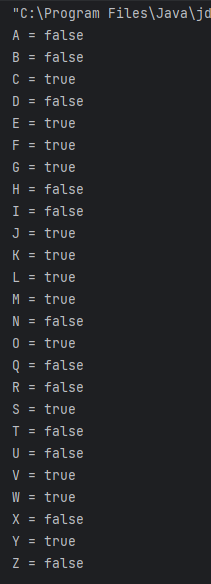
s) – (i + j) != -i + j t) i <= j && i >= c

u) i > 0 && j < 5 v) i > 0 || j < 5

w) x > y && i > 0 || j < 5 x) (3 \* i – 2 \* j) % (2 \* d – c) > 3 \* d

y) 2 \* ((i / 5) + (4 \* (j – 3)) % (i + j – 2)) >= 10 z) (i – 3 \* j) % (c + 2 \* d) / (x – y) >= 0

****



**Ejercicio 12.1:**

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

int x = 10;

System.out.println(x);

x++;

System.out.println(x);

System.out.println(++x);

System.out.println(x++);

System.out.println(x);

System.out.println(x++);

System.out.println(++x);

System.out.println(++x);

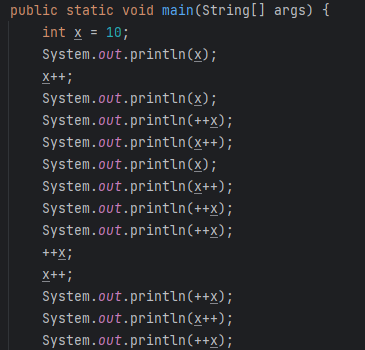
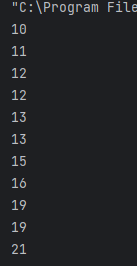
++x;

x++;

System.out.println(++x);

System.out.println(x++);

System.out.println(++x);



**Ejercicio 12.2:**

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

char A = 'c';

char B;

System.out.println(A++);

System.out.println(A++);

System.out.println(++A);

B = --A;

System.out.println(++A);

A++;

--B;

System.out.println(B++);

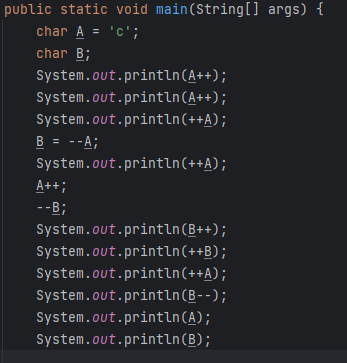
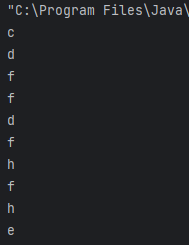
System.out.println(++B);

System.out.println(++A);

System.out.println(B--);

System.out.println(A);

System.out.println(B);



**Ejercicio 12.3:**

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.

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

a) i = ++j;

b) i = k++;

c) i = k + ++j;

d) i = i + j++;

e) i = j + ++k;

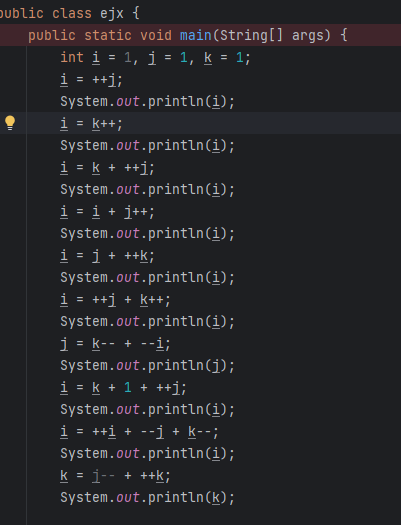
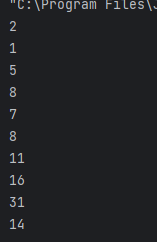
f) i = ++j + k++;

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

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

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

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



**Ejercicio 17:**

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

1.  El doble de un número N

N\*2

2.  El triple de un número N

N\*3

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% + M%

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

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’)

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. Con DN, MN, AN  día, mes y año de nacimiento de una persona y DA, MA, AA día, mes y año actual, escribe la expresión algorítmica que compruebe si tiene 18 años cumplidos.

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

**Ejercicio 18:**

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.

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;

16. Si el valor de una variable entera N es negativo, asignarle el valor 5 sino asignarle el valor 100.

N = N < 0 ? 5 : 100;