

Programación 1

Problemas 2

Expresiones



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza





Expresiones aritméticas

1. Expresión que determine el área de un rectángulo cuyos lados miden x e y .



Hoy, estamos interesados solo en expresiones...

$x * y$



...no en programas completos

```
#include <iostream>
using namespace std;

/* Pre:  ---
 * Post: Ha solicitado al usuario la longitud de los lados de un
 *        rectángulo y ha escrito en la pantalla el área del mismo.
 */
int main() {
    cout << "Introduzca las longitudes de los lados de un rectángulo: "
          << flush;
    double x, y;
    cin >> x >> y;

    cout << "El área del rectángulo es: " << x * y << endl;
    return 0;
}
```



Expresiones aritméticas

2. Expresión que determine el número de enteros comprendidos entre los enteros x e y , inclusive, siendo $x \leq y$.
3. Expresión que determine la longitud de una circunferencia de radio r .



Constante M_PI

- `<cmath>`
 - `const long double M_PI`
`= 3.14159265358979323846;`
- No forma parte del estándar de C++
 - Pero está disponible en los compiladores GNU GCC Compiler y en LLVM Clang Compiler para CentOS



Expresiones aritméticas

4. Expresión que determine cuántas bolas esféricas de radio 1,0 cm pueden almacenarse en una caja con forma de ortoedro (paralelepípedo ortogonal) cuyas dimensiones son x , y , z , todas ellas múltiplos de 10 cm.
5. Expresión que determine la longitud de hipotenusa de un triángulo rectángulo cuyos catetos miden a y b .



Función sqrt

- ❑ <cmath>
- ❑ **double sqrt (double x);**
 - Compute square root
 - Returns the *square root* of *x*.
- ❑ Parameters
 - *x*
 - ❑ Value whose square root is computed.
 - ❑ If the argument is negative, a *domain error* occurs.
- ❑ Return Value
 - ❑ Square root of *x*.
 - ❑ If *x* is negative, a *domain error* occurs:
- ❑ Fuente
 - <http://www.cplusplus.com/reference/cmath/sqrt/>



Expresiones aritméticas

6. Expresión que determine el valor de la suma de los enteros comprendidos en el intervalo $[x, y]$, siendo $x \leq y$.
7. Expresión que determine cuántos caramelos corresponden como máximo a cada uno de los n niños de un grupo, si repartimos un total de t caramelos entre ellos de modo equitativo.
8. Expresión que determine cuántos caramelos sobrarían tras hacer el reparto anterior.



Expresiones aritméticas con conversión de tipos

9. Expresión que determine el porcentaje de alumnos aprobados con respecto a los matriculados en una asignatura, siendo ***aprobados*** el número de alumnos aprobados y ***matriculados*** el número de alumnos matriculados en dicha asignatura. ¿Y la tasa de aprobados?
10. Expresión que redondee el valor de un dato real positivo ***x*** al entero más próximo.



Expresiones de relación y booleanas

11. Expresión cuyo valor sea *cierto* si y solo si el valor de **x** es mayor o igual que el valor de **y** .
12. Expresión cuyo valor sea *cierto* si y solo si el valor de **a** no es nulo.
13. Expresión cuyo valor sea *cierto* si y solo si los valores de **a** y **b** son iguales.



Expresiones de relación y booleanas

14. Expresión cuyo valor sea *cierto* si y solo si el valor de x es estrictamente mayor que la suma de los valores de a y b .
15. Expresión cuyo valor sea *cierto* si y solo si el valor de x está comprendido en el intervalo $[a, b]$, con $a \leq b$.
16. Expresión cuyo valor sea *cierto* si y solo si el valor de x está fuera del intervalo $[a, b]$, con $a \leq b$.



Expresiones de relación y booleanas

17. Expresión cuyo valor sea *cierto* si y solo si el valor de **x** es par, sabiendo que **$x \geq 0$** .
18. Expresión cuyo valor sea *cierto* si y solo si el valor de **b** es impar, sabiendo que **$b \geq 0$** .
19. Expresión cuyo valor sea *cierto* si y solo si el valor de **a** es múltiplo de 7, sabiendo que **$a \geq 0$** .



Expresiones de relación y booleanas

- 20. Expresión cuyo valor sea *cierto* si y solo si **d** es un entero par y positivo que divide al entero **n** .
- 21. Expresión cuyo valor sea *cierto* si y solo si un triángulo cuyos lados miden **a** , **b** y **c** , con **$a > b$** y **$b \geq c$** es un triángulo rectángulo.
- 22. Expresión cuyo valor sea *cierto* si y solo si puede existir un triángulo cuyos lados midan **a** , **b** y **c** , todos ellos valores positivos.



Expresiones con caracteres

- 23. Expresión cuyo valor sea *cierto* si y solo si **c** es un carácter cuyo valor es igual a 'A'.
- 24. Expresión cuyo valor sea *cierto* si y solo si **c** es un carácter cuyo valor es una letra mayúscula del alfabeto inglés.
- 25. Expresión que, suponiendo que el valor de **c** es una letra mayúscula del alfabeto inglés distinta a la 'Z', determine cuál es la letra que le sigue.



Soluciones

Necesita la declaración previa de `M_PI` en la biblioteca `<cmath>` o la siguiente declaración
`const double M_PI = 3.141592653589793;`

1. `x * y`
2. `2.0 * M_PI * r`
3. `y - x + 1`
4. `(x * y * z) / 8`
5. `sqrt(a * a + b * b)`
6. `(y - x + 1) * (x + y) / 2`
7. `t / n`
8. `t % n`
9. Porcentaje: `100.0 * aprobados / matriculados`
Tasa: **`double(aprobados) / double(matriculados)`**



Soluciones

10. `int(x + 0.5)`

11. `x >= y`

12. `a != 0`

13. `a == b`

14. `x > a + b`

15. `a <= x && x <= b`

16. `!(a <= x && x <= b)`

o, alternatively: `x < a || b < x`

17. `x % 2 == 0`



Soluciones

- 18. `b % 2 != 0`
- 19. `a % 7 == 0`
- 20. `(d % 2 == 0) && (d > 0) && (n % d == 0)`
- 21. `a * a == b * b + c * c`
- 22. `(a < b + c) || (b < a + c) || (c < a + b)`
- 23. `c == 'A'`
- 24. `('A' <= c) && (c <= 'Z')`
- 25. `char(int(c) + 1)`
o, simplemente: `c + 1`