**FUNCIONES**

**Funciones comunes.**

# 

# *1. Ejercicio*

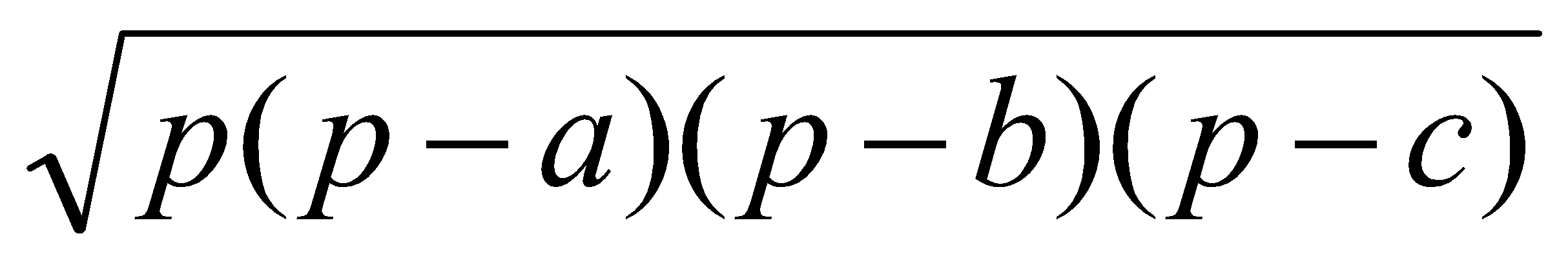
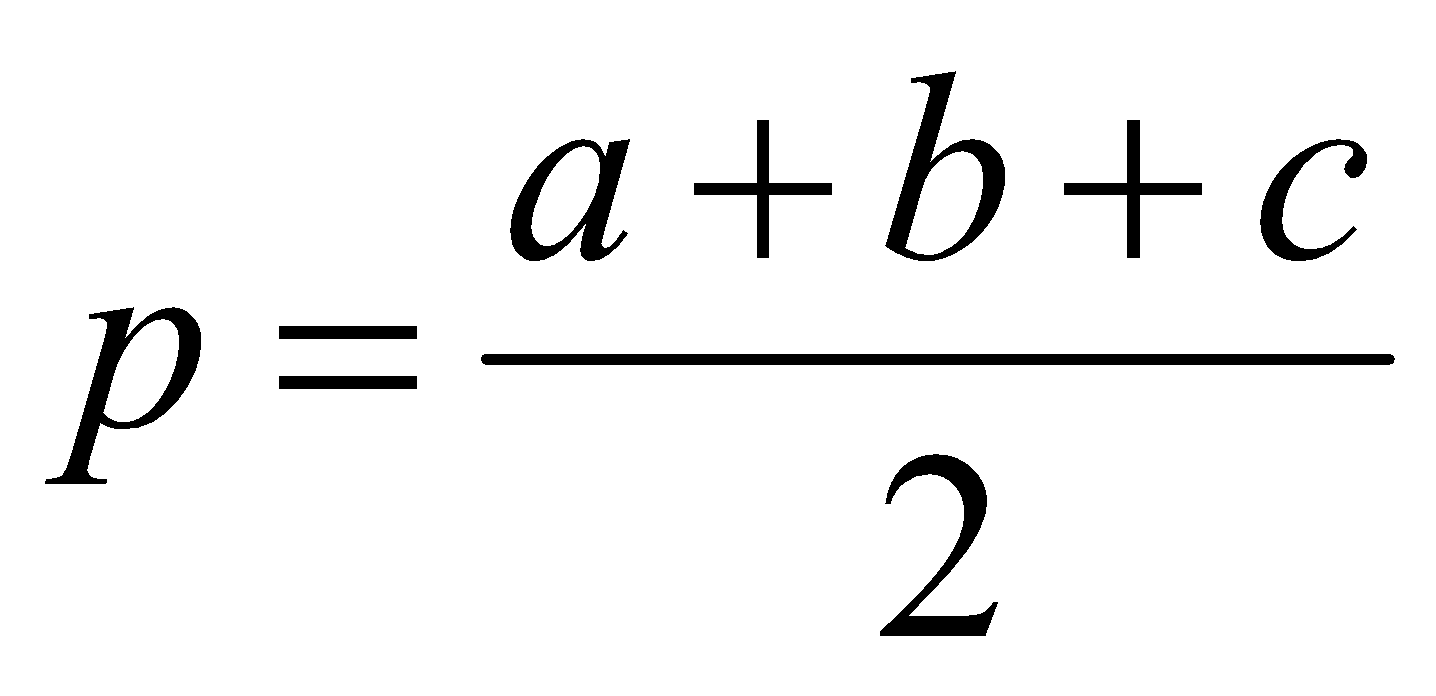
Escribir una función que reciba como parámetros la longitud de los lados de un triángulo y que **retorne** el área del mismo. Implementarla en un programa que reciba los parámetros como datos e informe el área calculada. Ejemplo:

Ingrese lado 1: ***2***

Ingrese lado 1: ***6***

Ingrese lado 1: ***7***

El área del triángulo es = 5.56

Ayuda: el área de cualquier triángulo equivale a  siendo 

# *2. Ejercicio*

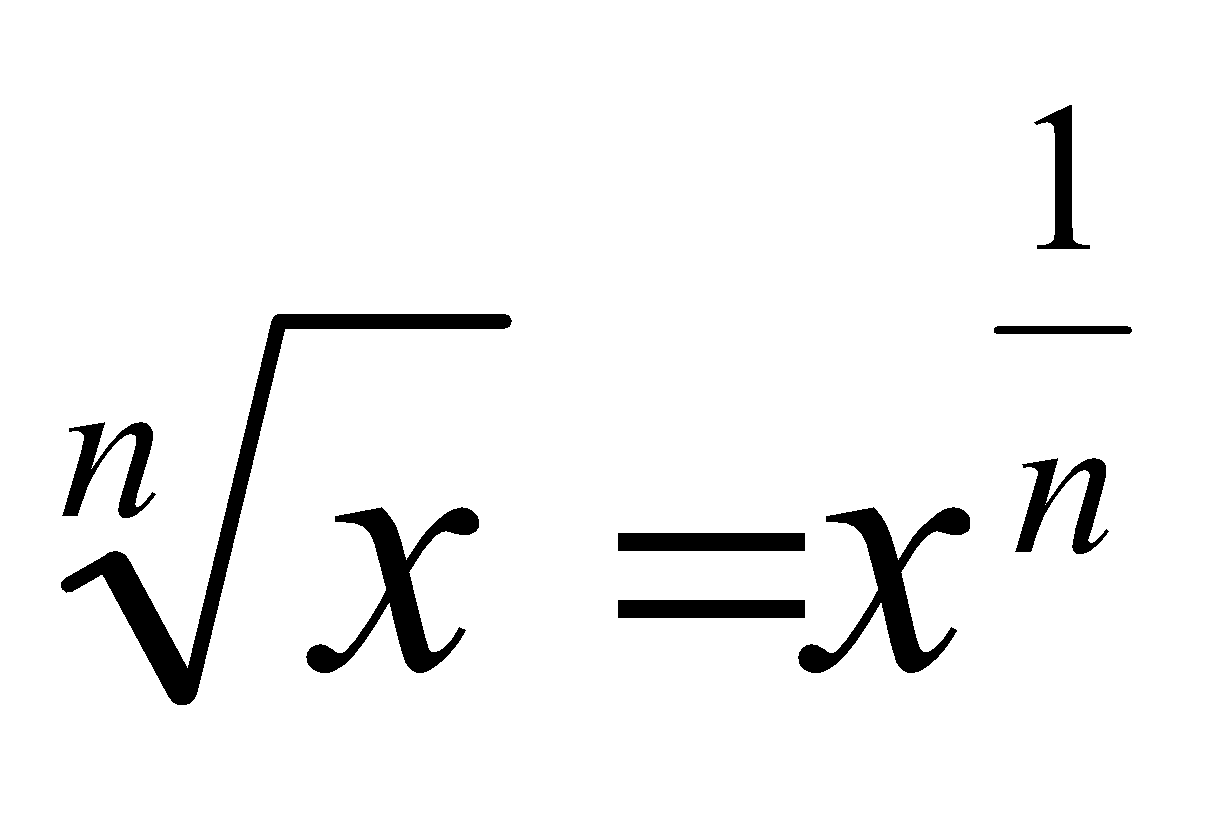
Programar una función que reciba como **parámetros un número real** ‘x’ y otro **entero positivo** ‘n’ mayor que 0. Esta función deberá **retornar** la raíz enésima de x. Utilizarla en un programa que solicite el número real, la raíz y luego imprima el valor calculado. Ejemplo:

Ingrese un número real: ***14***

Ingrese la raíz a calcular: ***3***

La raíz 3 de 14 es = 2.410142

Ayuda:

1. Sabiendo que , utilizar la función pow de la biblioteca math.h cuyo prototipo es ***double*** *pow(****double*** *base,* ***double*** *exponente);*
2. Tener en cuenta que no deben aparecer errores en tiempo de ejecución.

# *3. Ejercicio*

Confeccionar una función que reciba como parámetros un número entero positivo (en sistema decimal) y otro entero positivo entre 2 y 9. Luego, la función deberá retornar el valor recibido como primer parámetro representado en el sistema de numeración dado por el segundo parámetro. Aplicar dicha función a un programa que solicite el número decimal, la base a convertir y luego imprima el valor calculado. Ejemplo:

Ingrese el número decimal: ***527***

Ingrese la base: ***8***

527 (10) = 1017 (8)

# 

# *4. Ejercicio*

Hacer una función que reciba como parámetro un timestamp Unix y que imprima la hora a la que corresponde con el siguiente formato: *hora:minutos:segundos*. Por ejemplo, si se recibe como parámetro el timestamp 1468074007 debería imprimir 14:20:07. Implementarla en un programa que verifique su corrección.

# *5. Ejercicio*

Hacer una función sin parámetros que imprima por pantalla el mensaje “Buenos días”, “Buenas tardes” o “Buenas noches” según corresponda al horario actual del sistema. La decisión dependerá de la hora según el siguiente criterio: de 6:00 a 12:59 es día, de 13:00 a 19:59 es tarde y de 20:00 a 5:59 es noche. Implementarla en un programa que verifique su corrección.

**Funciones booleanas.**

# *6. Ejercicio*

Confeccionar una función booleana que determine si 2 números son coprimos. Dos números son coprimos o primos entre sí cuando sólo tienen como divisor común mayor el 1. Aplicarla en un programa en que se ingresen dos números enteros positivos y que informe si éstos son coprimos o no.

# *7. Ejercicio*

Confeccionar una función booleana que reciba como parámetros la longitud los **lados** de un triángulo (números reales) y determine si se trata o no de un triángulo rectángulo. Aplicarla en un programa en que se ingresen las 3 longitudes y que informe si se trata de un triángulo rectángulo o no.

# *8. Ejercicio*

Confeccionar una función booleana que reciba como parámetro un carácter y que determine si se trata de un separador. Se considera separador a todos aquellos símbolos o signos de puntuación que separen una palabra de otra. Por ejemplo: espacio, punto, coma, punto y coma, guión, barra, signo de exclamación, signo de interrogación (estos últimos, sólo de cierre), etc.

Aplicarla en un programa en que se ingresen caracteres sucesivamente hasta que se ingrese \*. Por cada carácter ingresado se debe informar si se trata de un separador o no.

# *9. Ejercicio*

Confeccionar **una** función booleana que reciba como parámetros 2 números enteros positivos y que informe si el primero de ellos es divisible por el segundo. Dentro de la función no se podrán usar estructuras que evalúen condiciones (if, switch, while, etc.).

Utilizarla en un programa en el que se ingresen dos números enteros y luego imprima dos mensajes: uno que indique si el primero es divisible por el segundo y otro que indique si el segundo es divisible por el primero.

# *10. Ejercicio*

Programar una función que reciba como parámetros dos números enteros positivos y que sea capaz de **retornar** un número al azar generado entre estos valores (ambos inclusive). Considere que la función debe aceptar como valor máximo un número mayor que el valor máximo capaz de ser generado por la función rand. No debe impedirse que existan valores dentro del rango que no puedan ser generados.

# *11. Ejercicio*

Programar una función que no reciba parámetros y que genere y **retorne** un número al azar dentro del rango que genera la función rand pero de manera tal que el promedio de la suma de las cifras de dicho número sea menor a 5. Por ejemplo, el número 8135 es válido siendo que 8+1+3+5 = 17, luego 17/4 = 4.25 y este número es menor que 5.

# *12. Ejercicio*

Programar una función que reciba como parámetros dos números enteros positivos y que sea capaz de **retornar** un número real al azar generado entre estos valores (ambos inclusive) con la particularidad de que dicho número generado debe tener como mucho, 2 cifras decimales.

**Funciones a implementar.**

En los siguientes ejercicios, construir las funciones cuyos prototipos se indican y de modo que el programa genere la salida solicitada. El código dado **no puede modificarse**, con lo cual únicamente puede escribirse la implementación de las funciones, respetando los prototipos.

# *13. Ejercicio*

El siguiente programa debe dibujar en pantalla un rectángulo de asteriscos.

#include <iostream>

using namespace std;

***void dibujar\_fila(unsigned int);***

***void dibujar\_rectangulo(unsigned int, unsigned int);***

int main()

{ unsigned int alto, ancho;

cout << "Ingrese ancho:";

cin >> ancho;

cout << "Ingrese altura:";

cin >> alto;

dibujar\_rectangulo(ancho, alto);

return 0;

}

# 

# *14. Ejercicio*

El siguiente programa permite ingresar un texto y luego informa la cantidad de letras del mismo.

#include <iostream>

using namespace std;

***unsigned int ingresar\_texto(void);***

***bool es\_letra(char);***

int main()

{ unsigned int len;

cout << "Ingrese texto (ENTER para finalizar): ";

len = ingresar\_texto();

cout << "El texto tiene "<< len << " letras." << endl;

return 0;

}

# 

# 

# *15. Ejercicio*

Este programa valida una fecha ingresada por teclado.

#include <iostream>

using namespace std;

***bool es\_bisiesto(unsigned int);***

***bool es\_fecha\_valida(unsigned int, unsigned int, unsigned int);***

int main()

{ unsigned int dia, mes anio;

cout << "Ingrese dia: ";

cin >> dia;

cout << "Ingrese mes: ";

cin >> mes;

cout << "Ingrese anio: ";

cin >> anio;

if(es\_fecha\_valida(dia, mes, anio))

{

cout <<"La fecha es valida" << endl;

}

else

{

cout << "La fecha NO es valida" << endl;

}

return 0;

}

**Funciones con números aleatorios.**

# *16. Ejercicio*

Programar una función que reciba como parámetros 2 números enteros y que **retorne** un número aleatorio entre estos 2 números (inclusive). Por ejemplo, si se invocara la función con parámetros 1 y 6, deberá devolver un número aleatorio entre 1 y 6 inclusive. Implementarla en un programa que solicite al usuario que ingrese el rango de valores aleatorios y la cantidad deseada de valor. A continuación imprimir la lista de números aleatorios según el criterio solicitado. Ejemplo:

Ingrese mínimo: ***1***

Ingrese máximo: ***20***

Ingrese cantidad de valores: ***10***

1, 8, 20, 14, 11, 17, 2, 1, 5, 6.

# *17. Ejercicio*

Programar una función que reciba como parámetros tres números enteros positivos y que sea capaz de **retornar** un número al azar generado entre los dos primeros valores (ambos inclusive) y que tenga, a lo sumo, tantos dígitos fraccionarios como indique el tercer parámetro. Ejemplo: azar(10, 20, 3); debe retornar un valor aleatorio entre 10 y 20 (ambos inclusive) que tenga hasta 3 dígitos aleatorios (p.e.: 10.234, 12.89, 20, etc.)