

# Ayudantía 2

`<cmath>` y funciones

Benjamín Aceituno

Programación Sección 16



# IMPORTANTE!!!

Mucho ojo con usar ChatGPT en las evaluaciones, además de copiar. Esto significará la nota mínima en la evaluación y si es muy grave pueden hasta reprobar la asignatura, mucho cuidado. (Se nota mucho cuando es ChatGPT y copia...)



## <cmath>

<cmath> es una librería al igual que <iostream>, esta nos permite importar una gran variedad de funciones matemáticas a nuestro programa, donde estas trabajan en números flotantes (float, double)

**pow (base X, exponente X):** Calcula la potencia de base elevado a exponente.

**sqrt (X):** Devuelve la raíz cuadrada de X.

**exp(X):** Calcula euler elevado a la potencia de X

**ceil(X):** Devuelve el entero más pequeño que es mayor o igual a X

**floor(X):** Devuelve el entero más grande que es menor o igual a X

**cos(X):** Calcula el coseno del ángulo X

**sin(X):** Calcula el seno del ángulo X

**tan(X):** Calcula la tangente del ángulo X

## Ejercicio

Realice un ejercicio en C++ que le permita transformar un número binario de 3 dígitos a un número decimal equivalente, use el siguiente método para transformar el número.

$$\begin{array}{ccccccccc} & & & 1 & 1 & 0 & 1 & 0 & 1 \\ & \swarrow & \searrow & \swarrow & \searrow & \swarrow & \searrow & \swarrow & \searrow \\ 1 \times 2^5 & + & 1 \times 2^4 & + & 0 \times 2^3 & + & 1 \times 2^2 & + & 0 \times 2^1 & + & 1 \times 2^0 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ 32 & + & 16 & + & 0 & + & 4 & + & 0 & + & 1 \\ \hline & & & & & & 53 & & & & \end{array}$$
$$110101_{(2)} = 53_{(10)}$$



# Solución

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

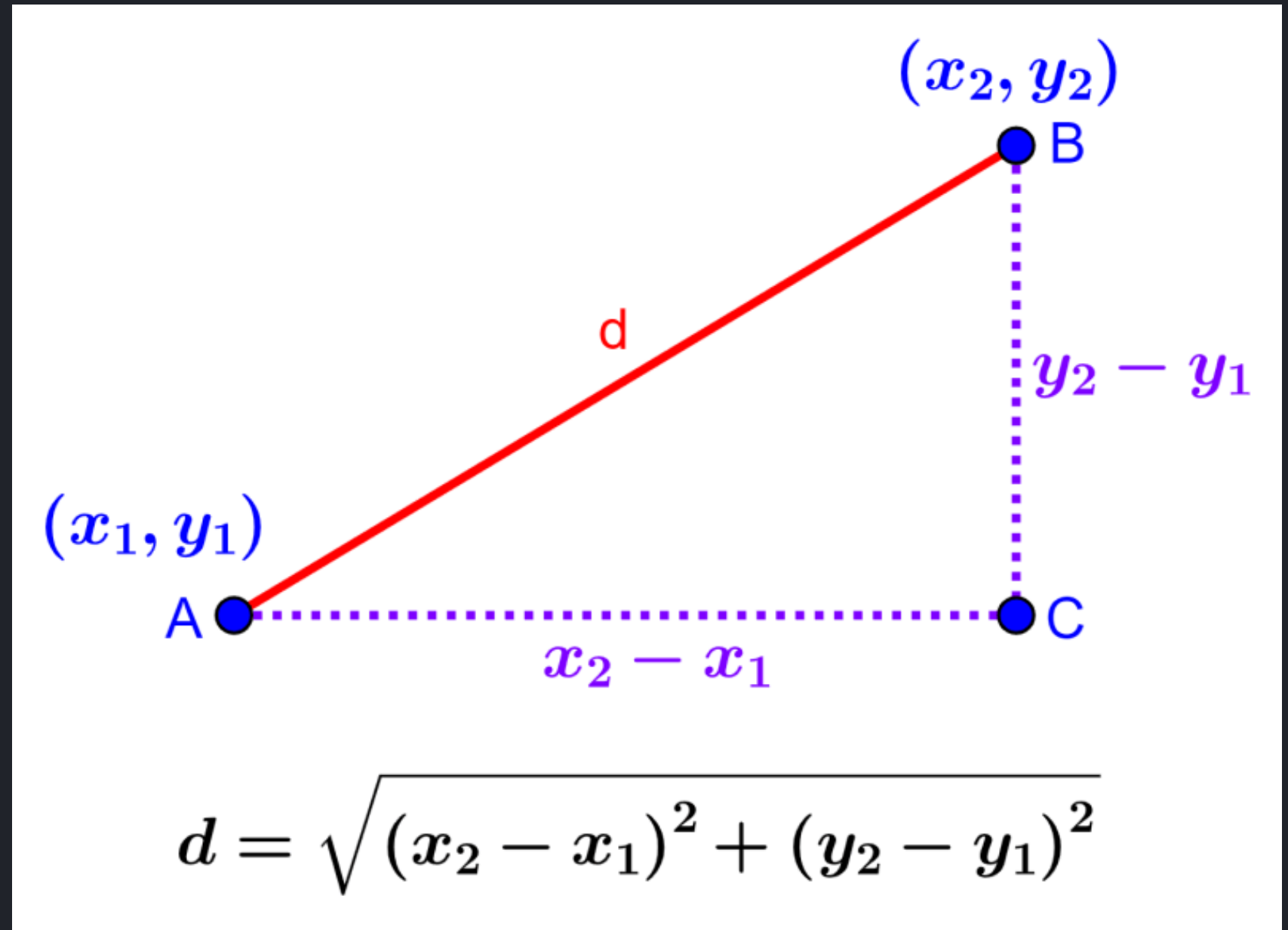
// Transformación de un número binario de 3 cifras a decimal.
int main()
{
    double primero, segundo, tercero, primeroD, segundoD, terceroD;
    // EJEMPLO
    cout << "Ingrese primer digito" << endl; // 1
    cin >> primero;
    cout << "Ingrese segundo digito" << endl; // 0
    cin >> segundo;
    cout << "Ingrese tercer digito" << endl; // 1
    cin >> tercero;

    primeroD = pow(2, 2) * primero; // 1 * 2*2
    segundoD = pow(2, 1) * segundo; // 0 * 2
    terceroD = pow(2, 0) * tercero; // 1 * 1
    // 4 + 0 + 1
    cout << "Su numero binario transformado a decimal es: " << primeroD + segundoD + terceroD << endl;

    return 0;
}
```

## Ejercicio

Realice un ejercicio en C++ que le permita saber la distancia entre dos puntos, ocupando la siguiente formula y las funciones sqrt y pow





# Solución

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    // Coordenadas del primer punto (x1, y1)
    double x1, y1;
    cout << "Ingrese las coordenadas del primer punto (x1 y1): ";
    cin >> x1 >> y1;

    // Coordenadas del segundo punto (x2, y2)
    double x2, y2;
    cout << "Ingrese las coordenadas del segundo punto (x2 y2): ";
    cin >> x2 >> y2;

    // Calcula las diferencias en las coordenadas
    double dx = x2 - x1;
    double dy = y2 - y1;

    // Calculo usando la formula del ejercicio
    double distancia = sqrt(pow(dx, 2) + pow(dy, 2));

    cout << "La distancia entre los dos puntos es: " << distancia << endl;

    return 0;
}
```

# Funciones propias

En C++ podemos crear nuestras propias funciones que realicen un bloque de código. Estas funciones reciben un argumento para realizar un retorno de un valor, de cierto modo, entregan un “resultado”

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

int areaRectangulo(int base, int altura){
    return base * altura;
}

int main(){
    int base, altura;
    cout << "Ingrese la base del rectangulo." << endl;
    cin >> base;
    cout << "Ingrese la altura del rectangulo." << endl;
    cin >> altura;
    int area = areaRectangulo(base, altura);
    cout << "El area del rectangulo es: " << area << endl;
}
```

La función tiene que ser definida, tener un tipo de variable (int) y retornar una variable del mismo tipo

Desde el int main “llamamos” a la función y le “pasamos” las variables



## Ejercicio

Realice la función `aplicarDescuento`. Esta función recibirá el precio y porcentaje de descuento que quiera el usuario y aplicará el descuento a un producto determinado. Esta función debe retornar el precio original del producto y restarle el descuento.

Ejemplo: `precioOriginal = 2000`, `descuento = 50`,  
`precioConDescuento = 1000`



iStock  
Credit: iridi

# Solución

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

double aplicarDescuento(double precioOriginal, double porcentajeDescuento) {
    double descuento = precioOriginal * (porcentajeDescuento / 100);
    return precioOriginal - descuento;
}

int main() {
    double precio, porcentaje;
    // Solicitar el precio original y el porcentaje de descuento
    cout << "Ingresa el precio original: ";
    cin >> precio;
    cout << "Ingresa el porcentaje de descuento: ";
    cin >> porcentaje;

    // Calcular el precio con el descuento aplicado
    double precioConDescuento = aplicarDescuento(precio, porcentaje);

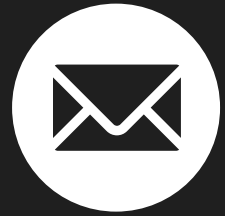
    // Mostrar el precio con el descuento
    cout << "El precio con el descuento aplicado es: " << precioConDescuento << endl;

    return 0;
}
```

# Contacto



+569 86031881



benjamin.aceituno@mail.udp.cl



benja.mp4



<https://github.com/benjamp4>

