

# ARRAYS

ricardo roque vilca

May 2025

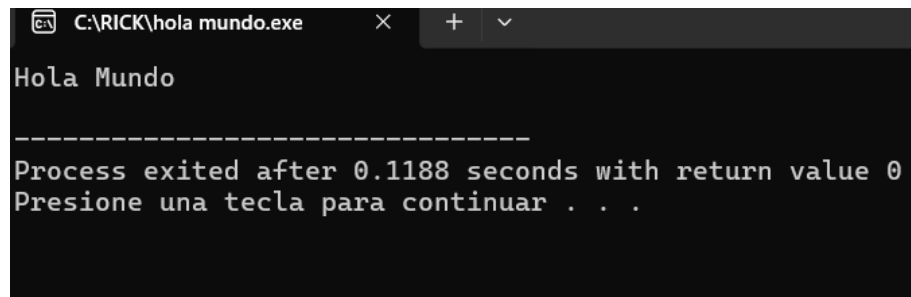
<https://github.com/RICARDORV23/ARRAYS.git>

## 1 HOLA MUNDO

```
#include <iostream>

using namespace std;

int main() {
    cout << "Hola-Mundo" << endl;
    return 0;
}
```



## 2 PROMEDIO DE 3 NUMEROS

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

int main() {
    double num1, num2, num3, suma, promedio;
```

```

    cout << "Introduce el primer n. mero:-";
    cin >> num1;

    cout << "Introduce el segundo n. mero:-";
    cin >> num2;

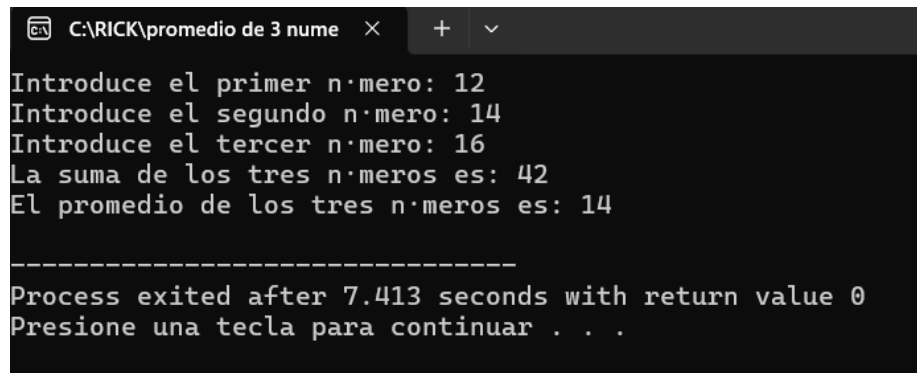
    cout << "Introduce el tercer n. mero:-";
    cin >> num3;

    suma = num1 + num2 + num3;
    promedio = suma / 3;

    cout << "La suma de los tres n. meros es:-" << suma << endl;
    cout << "El promedio de los tres n. meros es:-" << promedio << endl;

    return 0;
}

```



```

C:\RICK\promedio de 3 nume
Introduce el primer n. mero: 12
Introduce el segundo n. mero: 14
Introduce el tercer n. mero: 16
La suma de los tres n. meros es: 42
El promedio de los tres n. meros es: 14

-----
Process exited after 7.413 seconds with return value 0
Presione una tecla para continuar . . .

```

### 3 AREA DE UN CIRCULO

```

#include <iostream>
#include <cmath>

using namespace std;

int main() {
    double radio, area;

    cout << "Introduce el radio del c. rculo:-";
    cin >> radio;
}

```

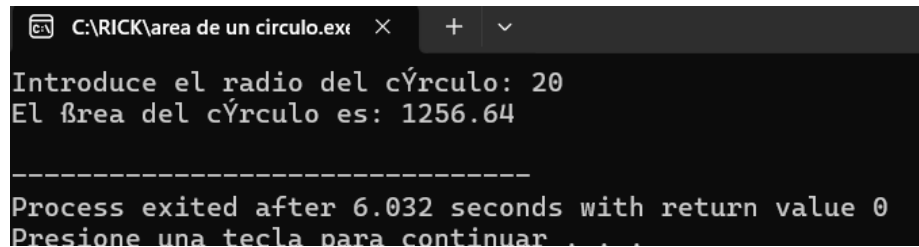
```

    area = M_PI * pow(radio, 2);

    cout << "El área del círculo es:" << area << endl;

    return 0;
}

```



```

C:\RICK\area de un circulo.exe
Introduce el radio del círculo: 20
El área del círculo es: 1256.64

-----
Process exited after 6.032 seconds with return value 0
Presione una tecla para continuar...

```

## 4 CALIFICACION DE 30 ESTUDIANTES

Listing 1: Programa para calcular el promedio de calificaciones.

```

#include <iostream>

using namespace std;

int main() {
    const int NUM_ESTUDIANTES = 30;
    double calificaciones[NUM_ESTUDIANTES];
    double suma = 0, promedio;
    int estudiantesArribaDelPromedio = 0;

    cout << "Introduce las calificaciones de los " <<
        NUM_ESTUDIANTES << " estudiantes:\n";
    for (int i = 0; i < NUM_ESTUDIANTES; i++) {
        cout << "Estudiante " << (i + 1) << ": ";
        cin >> calificaciones[i];
        suma += calificaciones[i];
    }

    promedio = suma / NUM_ESTUDIANTES;

    for (int i = 0; i < NUM_ESTUDIANTES; i++) {
        if (calificaciones[i] > promedio) {
            estudiantesArribaDelPromedio++;
        }
    }
}

```

```

        cout << "\nEl promedio general de las calificaciones es:
        " << promedio << endl;
        cout << "Numero de estudiantes por encima del promedio:
        " << estudiantesArribaDelPromedio << endl;

        return 0;
    }

```

## 5 NUMEROS PRIMOS

Listing 2: Programa para calcular el promedio de calificaciones.

```

#include <iostream>

using namespace std;

int main() {
    const int NUM_ESTUDIANTES = 30;
    double calificaciones[NUM_ESTUDIANTES];
    double suma = 0, promedio;
    int estudiantesArribaDelPromedio = 0;

    // Leer las calificaciones de los estudiantes
    cout << "Introduce las calificaciones de los " <<
        NUM_ESTUDIANTES << " estudiantes:\n";
    for (int i = 0; i < NUM_ESTUDIANTES; i++) {
        cout << "Estudiante " << (i + 1) << ": ";
        cin >> calificaciones[i];
        suma += calificaciones[i];
    }

    promedio = suma / NUM_ESTUDIANTES;

    for (int i = 0; i < NUM_ESTUDIANTES; i++) {
        if (calificaciones[i] > promedio) {
            estudiantesArribaDelPromedio++;
        }
    }

    cout << "\nEl promedio general de las calificaciones es:
    " << promedio << endl;
    cout << "Numero de estudiantes por encima del promedio:
    " << estudiantesArribaDelPromedio << endl;

    return 0;
}

```

## 6 INGRESA 10 NUMEROS ENTEROS

Listing 3: Programa para rotar un arreglo de enteros.

```
#include <iostream>

using namespace std;

int main() {
    const int TAMANO = 10;
    int numeros[TAMANO];

    cout << "Ingresa 10 n meros enteros:" << endl;
    for (int i = 0; i < TAMANO; ++i) {
        cout << "N mero " << (i + 1) << ": ";
        cin >> numeros[i];
    }

    int ultimo = numeros[TAMANO - 1];

    for (int i = TAMANO - 2; i >= 0; --i) {
        numeros[i + 1] = numeros[i];
    }

    numeros[0] = ultimo;

    cout << "Arreglo despu s de rotar una posici n hacia la derecha:" << endl;
    for (int i = 0; i < TAMANO; ++i) {
        cout << numeros[i] << " ";
    }
    cout << endl;

    return 0;
}
```

## 7 ALAMCEN DE 20 NUMEROS

Listing 4: Programa para contar la frecuencia de números en un arreglo.

```
#include <iostream>

using namespace std;

int main() {
    const int TAMANO = 20;
    int numeros[TAMANO];
    bool yaContado[TAMANO] = {false};
```

```

    cout << "Ingresa 20 n meros enteros:" << endl;
    for (int i = 0; i < TAMANO; ++i) {
        cout << "N mero " << (i + 1) << ": ";
        cin >> numeros[i];
    }

    for (int i = 0; i < TAMANO; ++i) {
        if (!yaContado[i]) {
            int frecuencia = 1;
            for (int j = i + 1; j < TAMANO; ++j) {
                if (numeros[i] == numeros[j]) {
                    frecuencia++;
                    yaContado[j] = true;
                }
            }
            cout << "El n mero " << numeros[i] << " aparece " << frecuencia << " vez/veces." << endl;
        }
    }

    return 0;
}

```

## 8 10 ELEMENTOS

Listing 5: Programa para comparar elementos en dos arreglos.

```

#include <iostream>

using namespace std;

int main() {
    const int TAMANO = 10;
    int arreglo1[TAMANO], arreglo2[TAMANO];
    int contadorIguales = 0;

    cout << "Ingresa 10 elementos para el primer arreglo:" << endl;
    for (int i = 0; i < TAMANO; ++i) {
        cout << "Elemento " << (i + 1) << ": ";
        cin >> arreglo1[i];
    }

    cout << "Ingresa 10 elementos para el segundo arreglo:" << endl;
    for (int i = 0; i < TAMANO; ++i) {
        cout << "Elemento " << (i + 1) << ": ";
    }
}

```

```

        cin >> arreglo2[i];
    }

    for (int i = 0; i < TAMANO; ++i) {
        if (arreglo1[i] == arreglo2[i]) {
            contadorIguales++;
        }
    }

    cout << "El n mero de elementos iguales en las mismas
    posiciones es:" << contadorIguales << endl;

    return 0;
}

```