

# Trabajo en clase

**Nombre:** Juan David Torres Avila

**Asignatura:** Programación II (IS284)

**Grupo:** 5°

## Ejercicio 1: Calculadora Simple

**Descripción del Problema:** Se necesita un programa que realice operaciones aritméticas simples sobre dos números enteros. El programa debe permitir al usuario seleccionar la operación deseada.

**Formato de Entrada:** El programa solicitará al usuario dos números enteros y luego un carácter representando la operación deseada (+, -, \*, /).

**Formato de Salida:** El programa imprimirá el resultado de la operación aritmética.

### Código:

```
#include <iostream>

using namespace std;
// Ejercicio 1: Calculadora Simple
int main() {
    int num1, num2;
    char operation;

    cout << "Ingrese el primer número: ";
    cin >> num1;

    cout << "Ingrese el segundo número: ";
    cin >> num2;

    cout << "Ingrese la operación (+, -, *, /): ";
    cin >> operation;

    int result;
    if (operation == '+') {
        result = num1 + num2;
    } else if (operation == '-') {
        result = num1 - num2;
    } else if (operation == '*') {
```

```

    result = num1 * num2;
} else if (operation == '/') {
    if (num2 == 0) {
        cout << "Error: No se puede dividir por cero." << endl;
        return 1;
    }
    result = num1 / num2;
} else {
    cout << "Error: Operación inválida." << endl;
    return 1;
}

cout << "Resultado: " << result << endl;

return 0;
}

```

### Ejemplo de resultado esperado:

```

Ingrese el primer número: 44
Ingrese el segundo número: 89
Ingrese la operación (+, -, *, /): -
Resultado: -45

```

## Ejercicio 2: Clasificador de Edades

**Descripción del Problema:** Se solicita un programa que clasifique a una persona en una de tres categorías: Niño, Adolescente o Adulto, según su edad.

**Formato de Entrada:** El programa solicitará al usuario su edad como un número entero positivo.

**Formato de Salida:** El programa imprimirá la categoría correspondiente: "Niño", "Adolescente" o "Adulto".

### Código:

```
#include <iostream>
```

```
using namespace std;
```

```
// Ejercicio 2: Clasificador de Edades
```

```
int main() {
```

```
int age;

cout << "Ingrese su edad en años: ";
cin >> age;

if (age < 1 || age > 100) {
    cout << "Edad inválida." << endl;
    return 1;
}

if (age <= 9) {
    cout << "Niño" << endl;
} else if (age <= 19) {
    cout << "Adolescente" << endl;
} else {
    cout << "Adulto" << endl;
}

return 0;
}
```

**Ejemplo de resultado esperado:**

```
Ingrese su edad en años: 19
Adolescente
```

### Ejercicio 3: Ordenador de Números

**Descripción del Problema:** Se necesita un programa que ordene tres números enteros distintos de menor a mayor.

**Formato de Entrada:** El programa solicitará al usuario tres números enteros distintos.

**Formato de Salida:** El programa imprimirá los tres números ordenados de menor a mayor.

#### Código:

```
#include <iostream>

using namespace std;
// Ejercicio 3: Ordenador de Numeros
int main() {
    int num1, num2, num3;

    cout << "Ingrese el primer número: ";
    cin >> num1;

    cout << "Ingrese el segundo número: ";
    cin >> num2;

    cout << "Ingrese el tercer número: ";
    cin >> num3;

    if (num1 == num2 or num3 == num2 or num1 == num3 ) {
        cout << "Hay números iguales, ingrese números diferentes." << endl;
        return 1;}

    if (num1 > num2) {
        swap(num1, num2);
    }
    if (num1 > num3) {
        swap(num1, num3);
    }
    if (num2 > num3) {
        swap(num2, num3);
    }

    cout << num1 << " " << num2 << " " << num3 << endl;
```

```
return 0;  
}
```

#### Ejemplo de resultado esperado:

```
Ingrese el primer número: 5  
Ingrese el segundo número: 3  
Ingrese el tercer número: 88  
3 5 88
```

## 4. Intersección de intervalos

### Descripción

Dados dos intervalos cerrados  $y$  , escribe un programa que determine si existe algún número que pertenezca a ambos intervalos.

### Entrada

Cuatro enteros  $a$  ,  $b$  ,  $c$  , y  $d$  . Puedes suponer que  $a \leq b$  y  $c \leq d$  .

### Salida

Un 1 si existe algún número que pertenezca a ambos intervalos y un 0 en otro caso.

### Código:

```
#include <iostream>
```

```
using namespace std;  
//Intersección de intervalos  
int intersection(int a, int b, int c, int d) {  
    if (b < c || a > d) {  
        return 0;  
    } else {  
        return 1;  
    }  
}
```

```
int main() {  
    int a, b, c, d;  
  
    cout << "Ingrese el límite inferior del primer intervalo: ";  
    cin >> a;  
  
    cout << "Ingrese el límite superior del primer intervalo: ";  
    cin >> b;
```

```

cout << "Ingrese el límite inferior del segundo intervalo: ";
cin >> c;

cout << "Ingrese el límite superior del segundo intervalo: ";
cin >> d;

int resultado = intersection(a, b, c, d);

if (resultado == 1) {
    cout << "Los intervalos se intersectan." << endl;
} else {
    cout << "Los intervalos no se intersectan." << endl;
}

return 0;
}

```

#### Ejemplo de resultado esperado:

```

Ingrese el límite inferior del primer intervalo: 14
Ingrese el límite superior del primer intervalo: 44
Ingrese el límite inferior del segundo intervalo: 55
Ingrese el límite superior del segundo intervalo: 92
Los intervalos no se intersectan.

```

## 5. Dos gatos y un ratón

### Descripción

Dos gatos y un ratón se encuentran en varias posiciones sobre una línea. Como valor de entrada te vamos a dar sus posiciones iniciales. Tu tarea es determinar cuál gato alcanzará al ratón primero, asumiendo que el ratón no se mueve y ambos gatos se mueven a la misma velocidad.

OJO: Si ambos gatos llegan al mismo tiempo, el ratón podrá escapar mientras que los dos gatos pelean.

### Entrada

Se te darán como valores de entrada tres enteros , y separados por un espacio, los cuales representan respectivamente las posiciones del gato A, del gato B y del ratón C. Puedes suponer que .

### Salida

Si el gato A alcanza primero al ratón, imprimir como salida "gato A".  
Si el gato B alcanza primero al ratón, imprimir como salida "gato B".  
Si ambos gatos alcanzan al ratón C al mismo tiempo, imprimir como salida "ratón C".

### Código:

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

int main() {
    int gatoA, gatoB, ratonC;

    cout << "Ingrese la posición de los gatos y el raton: " << endl << "Gato A: ";
    cin >> gatoA;
    cout << "Gato B: ";
    cin >> gatoB;
    cout << "Raton C: ";
    cin >> ratonC;

    if ((gatoA < 1 || gatoA > 100) or (gatoB < 1 || gatoB > 100) or (ratonC < 1 || ratonC > 100)) {
        cout << "Los valores deben estar entre 1 y 100." << endl;
        return 1;}
    if (abs(gatoA - ratonC) < abs(gatoB - ratonC)) {
        cout << "gato A" << endl;
    } else if (abs(gatoA - ratonC) > abs(gatoB - ratonC)) {
        cout << "gato B" << endl;
    } else {
        cout << "raton C" << endl;
    }

    return 0;
}
```

### Ejemplo de resultado esperado:

```
Ingrese la posición de los gatos y el raton:
Gato A: 5
Gato B: 3
Raton C: 19
gato A
```