

Universidad Mariano Gálvez de Guatemala Facultad de Ingeniería Carrera de Ingeniería en
Sistemas de Información

Programación 1

Sección “A”



Laboratorio 4

Abner Aaron Muñoz Xoy – 9959-25-13959

Guatemala, 20 de enero 2025

4.11

4.11 (Corrija los errores de código) Identifique y corrija los errores en cada uno de los siguientes fragmentos de código:

```
a) if ( edad >= 65 );
    cout << "Edad es mayor o igual que 65" << endl;
    else
    cout << "Edad es menor que 65" << endl;

b) if ( edad >= 65 )
    cout << "Edad es mayor o igual que 65" << endl;
    else;
    cout << "Edad es menor que 65" << endl;

c) unsigned int x = 1;
    unsigned int total;

    while ( x <= 10 )
    {
        total += x;
        ++x;
    }

d) while ( x <= 100 )
    total += x;
    ++x;

e) while ( y > 0 )
    {
        cout << y << endl;
        ++y;
    }
```

Inciso a y b

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

int main() {
    int edad;

    cout << "Ingrese la edad: ";
    cin >> edad;

    if (edad >= 65)
        cout << "Edad es mayor o igual que 65" << endl;
    else
        cout << "Edad es menor que 65" << endl;

    return 0;
}
```

```
C:\Users\IntecapAbner x + - □ ×
Ingrese la edad: 66
Edad es mayor o igual que 65

Process returned 0 (0x0) execution time : 1
.871 s
Press any key to continue.
```

Inciso c y d

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

int main() {
    unsigned int x = 1;
    unsigned int total = 0;

    while (x <= 10) {
        total += x;
        x++;
    }
    cout << "Total: " << total << endl;

    return 0;
}
```

```
C:\Users\IntecapAbner\Docu x +
Total: 55

Process returned 0 (0x0) e
Press any key to continue.
```

e)

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

int main() {
    int y = 1;

    while (y > 0 && y <= 10) {
        cout << y << endl;
        y++;
    }

    return 0;
}
```

4.13

4.13 (Kilometraje de gasolina) Los conductores se preocupan acerca del kilometraje de sus automóviles. Un conductor ha llevado el registro de varios viajes, anotando los kilómetros conducidos y los litros usados en cada viaje. Desarrolle un programa en C++ que utilice una instrucción `while` para recibir como entrada los kilómetros conducidos y los litros usados por cada viaje, y que imprima el total de kilómetros por litro obtenidos en todos los reabastecimientos hasta este punto.

```

Escriba los kilometros usados (-1 para salir): 287
Escriba los litros: 13
KPL en este reabastecimiento: 22.076923
Total KPL: 22.076923

Escriba los kilometros usados (-1 para salir): 200
Escriba los litros: 10
KPL en este reabastecimiento: 20.000000
Total KPL: 21.173913

Escriba los kilometros usados (-1 para salir): 120
Escriba los litros: 5
KPL en este reabastecimiento: 24.000000
Total KPL: 21.678571

Escriba los kilometros usados (-1 para salir): -1

```

```

#include <iostream>
using namespace std;

int main(){
    double kilometros, litros;
    double totalKm = 0, totalLitros = 0;

    cout << "Ingrese los kilometros usados (-1 para salir): ";
    cin >> kilometros;

    while (kilometros != -1){
        cout << "Escriba los litros: ";
        cin >> litros;

        cout << "KPL en este reabastecimiento: " << kilometros / litros << endl;

        totalKm += kilometros;
        totalLitros += litros;
    }
    if (totalLitros != 0)
        cout << "Total KPL: " << totalKm / totalLitros << endl;
    return 0;
}

```

```

Ingrese los kilometros usados (-1 para salir):
15
Escriba los litros: 45
KPL en este reabastecimiento: 0.333333
Escriba los litros: 5
KPL en este reabastecimiento: 3
Escriba los litros: 3
KPL en este reabastecimiento: 5
Escriba los litros: -1
KPL en este reabastecimiento: -15
Escriba los litros:

```

4.15

4.15 (Calculadora de comisiones de ventas) Una empresa grande paga a sus vendedores mediante comisiones. Los vendedores reciben \$200 por semana, más el 9% de sus ventas brutas durante esa semana. Por ejemplo, un vendedor que vende \$5000 de productos químicos en una semana, recibe \$200 más el 9% de \$5000, o un total de \$650. Desarrolle un programa en C++ que utilice una instrucción `while` para recibir como entrada las ventas

Ejercicios 151

rutas de cada vendedor de la semana anterior, y que calcule y muestre los ingresos de ese vendedor. Procese las cifras de un vendedor a la vez.

```

Introduzca las ventas en dolares (-1 para salir): 5000.00
El salario es: $650.00

Introduzca las ventas en dolares (-1 para salir): 6000.00
El salario es: $740.00

Introduzca las ventas en dolares (-1 para salir): 7000.00
El salario es: $830.00

Introduzca las ventas en dolares (-1 para salir): -1

```

```

#include <iostream>
using namespace std;

int main(){
    double ventas;
    double totalKm = 0, totalLitros = 0;

    cout << "Introduzca las ventas en dolares (-1 para salir): ";
    cin >> ventas;

    while (ventas != -1){
        double salario = 200 + (ventas * 0.09);

        cout << "El salario es: $ " << salario << endl;

        cout << "Introduzca las ventas en dolares (-1 para salir): ";
        cin >> ventas;
    }

    return 0;
}

```

```

C:\Users\IntecapAbner\Docu x + v - □
Introduzca las ventas en dolares (-1 para salir): 30000
El salario es: $ 2900
Introduzca las ventas en dolares (-1 para salir): -1
Process returned 0 (0x0) execution time : 10.023 s
Press any key to continue.

```

4.17

4.17 (Encontrar el más grande) El proceso de encontrar el número más grande (es decir, el máximo de un grupo de números) se utiliza frecuentemente en aplicaciones de computadora. Por ejemplo, un programa para determinar el ganador de un concurso de ventas recibe como entrada el número de unidades vendidas por cada vendedor. El vendedor que haya vendido más unidades es el que gana el concurso. Escriba un programa en pseudocódigo y después una aplicación en C++ que utilice una instrucción `while` para determinar e imprimir el mayor número de una serie de 10 números introducidos por el usuario. Su programa debe utilizar tres variables, como se muestra a continuación:

contador:	Un contador para contar hasta 10 (es decir, para llevar el registro de cuántos números se han introducido, y para detectar cuando se hayan procesado los 10 números).
numero:	El número actual que se introduce al programa.
mayor:	El número más grande encontrado hasta ahora.

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

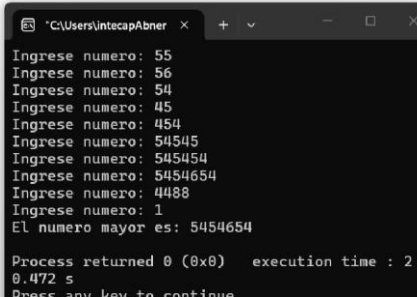
int main(){
    int numero;
    int mayor = 0;
    int contador = 1;

    while (contador <= 10){
        cout << "Ingrese numero: ";
        cin >> numero;

        if (contador == 1 || numero > mayor)
            mayor = numero;
        contador++;
    }

    cout << "El numero mayor es: " << mayor << endl;

    return 0;
}
```



4.19

4.19 (Encontrar los dos números más grandes) Utilizando una metodología similar a la del ejercicio 4.17, encuentre los *dos* valores más grandes de los 10 que se introdujeron. [Nota: debe introducir cada número sólo una vez].

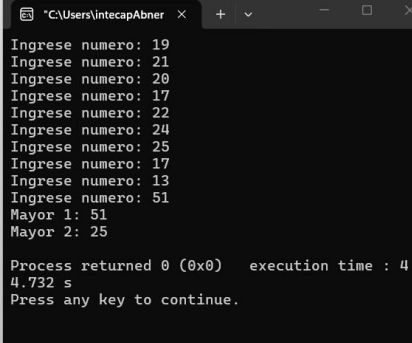
```
#include <iostream>
using namespace std;

int main(){
    int numero;
    int mayor1 = 0, mayor2 = 0;
    int contador = 1;

    while (contador <= 10){
        cout << "Ingrese numero: ";
        cin >> numero;
        if (numero > mayor1){
            mayor2 = mayor1;
            mayor1 = numero;
        } else if (numero > mayor2){
            mayor2 = numero;
        }
        contador++;
    }

    cout << "Mayor 1: " << mayor1 << endl;
    cout << "Mayor 2: " << mayor2 << endl;

    return 0;
}
```



4.21 : Omprime 10 filas intercaladas de * y +.

4.21 ¿Qué hace este programa? ¿Qué es lo que imprime el siguiente programa?

```
1 // Ejercicio 4.21: ej04_21.cpp
2 // ¿Qué es lo que imprime este programa?
3 #include <iostream>
4 using namespace std;
5
6 int main()
7 {
8     unsigned int cuenta = 1; // inicializa cuenta
9
10    while ( cuenta <= 10 ) // itera 10 veces
11    {
12        // imprime una línea de texto
13        cout << ( cuenta % 2 ? "*****" : "+++++" ) << endl;
14        ++cuenta; // incrementa cuenta
15    } // fin de while
16 } // fin de main
```

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

int main() {
    unsigned int cuenta = 1;

    while (cuenta <= 10) {
        cout << (cuenta % 2 ? "*****" : "+++++") << endl;
        cuenta++;
    }

    return 0;
}
```

```
****
++++
****
++++
****
++++
****
++++
****
++++

Process returned 0 (0x0)   execution time : 0
.166 s
Press any key to continue.
```

4.23 (Problema del `else` suelto) Determine la salida de cada uno de los siguientes conjuntos de código, cuando `x` es 11 y `y` es 9. El compilador ignora la sangría en un programa en C++. El compilador de C++ siempre asocia un `else` con el `if` anterior, a menos que se le indique de otra forma mediante la colocación de llaves (`{}`). A primera vista, el programador tal vez no esté seguro de cuál `if` corresponde a cuál `else`; esta situación se conoce como el

“problema del `else` suelto”. Hemos eliminado la sangría del siguiente código para hacer el problema más retador. [Sugerencia: aplique las convenciones de sangría que aprendió].

```
a) if ( x < 10 )
    if ( y > 10 )
        cout << "*****" << endl;
    else
        cout << "#####" << endl;
    cout << "SSSSS" << endl;
b) if ( x < 10 )
    {
        if ( y > 10 )
            cout << "*****" << endl;
    }
    else
    {
        cout << "#####" << endl;
        cout << "SSSSS" << endl;
    }
```

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

int main() {
    int x = 11;
    int y = 9;

    if (x < 20) {
        if (y > 10)
            cout << "*****" << endl;
        else
            cout << "#####" << endl;
    }
    cout << "$$$$$" << endl;

    return 0;
}
```

```
####
$$$$

Process returned 0 (0x0)   execution time : 0
.150 s
Press any key to continue.
```

4.25 **4.25 (Cuadrado de asteriscos)** Escriba un programa que pida al usuario que introduzca el tamaño del lado de un cuadrado y que muestre un cuadrado hueco de ese tamaño, compuesto de asteriscos y espacios en blanco. Su programa debe funcionar con cuadrados que tengan lados de todas las longitudes entre 1 y 20. Por ejemplo, si su programa lee un tamaño de 5, debe imprimir

```
*****
*   *
*   *
*   *
*   *
*****
```

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

int main() {
    int lado;

    cout << "Ingrese el tamaño del lado: ";
    cin >> lado;

    for (int i = 1; i <= lado; i++) {
        for (int j = 1; j <= lado; j++) {
            if (i == 1 || i == lado || j == 1 || j == lado)
                cout << "*";
            else
                cout << " ";
        }
        cout << "\n";
    }

    return 0;
}
```

```
Ingrese el tamaño del lado: 10
***** * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Process returned 0 (0x0)   execution time : 2
.847 s
Press any key to continue.
```

4.27 **4.27 (Imprimir el equivalente decimal de un número binario)** Escriba un programa que reciba como entrada un entero que contenga sólo ceros y unos (es decir, un entero "binario"), y que imprima su equivalente decimal. Use los operadores módulo y división para elegir los dígitos del número "binario" uno a la vez, de derecha a izquierda. En forma parecida al sistema numérico decimal, en donde el dígito más a la derecha tiene un valor posicional de 1 y el siguiente dígito a la izquierda tiene un valor posicional de 10, después 100, después 1000, etcétera, en el sistema numérico binario, el dígito más a la derecha tiene un valor posicional de 1, el siguiente dígito a la izquierda tiene un valor posicional de 2, luego 4, luego 8, etcétera. Así, el número decimal 234 se puede interpretar como $2 * 100 + 3 * 10 + 4 * 1$. El equivalente decimal del número binario 1101 es $1 * 1 + 0 * 2 + 1 * 4 + 1 * 8$, o $1 + 0 + 4 + 8$, o 13. [Nota: para aprender más acerca de los números binarios, consulte el apéndice D].


```

#include <iostream>
using namespace std;

int main() {
    int binario;
    int decimal = 0;
    int base = 1;
    int digito;

    cout << "Ingrse numero binario: ";
    cin >> binario;

    while (binario > 0) {
        digito = binario % 10;
        decimal += digito * base;
        base = base * 2;
        binario = binario / 10;
    }
    cout << "Numero decimal: " << decimal << endl;

    return 0;
}

```

```

C:\Users\IntecapAbner
Ingrse numero binario: 0100
Numero decimal: 4

Process returned 0 (0x0)   execution time : 0.283 s
Press any key to continue.

```

4.29

4.29 (*Múltiplos de 2 con un ciclo infinito*) Escriba un programa que imprima las potencias del entero 2; a saber, 2, 4, 8, 16, 32, 64, etcétera. Su ciclo `while` no debe terminar (es decir, debe crear un ciclo infinito). Para ello, simplemente use la palabra clave `true` como la expresión para la instrucción `while`. ¿Qué ocurre cuando ejecuta este programa?

```

#include <iostream>
using namespace std;

int main() {
    int numero = 2;

    while (true) {
        cout << numero << endl;
        numero = numero * 2;
    }

    return 0;
}

```

4.31

4.31 ¿Qué está mal con la siguiente instrucción? Proporcione la instrucción correcta para realizar lo que probablemente el programador trataba de hacer.

```
cout << ++( x + y );
```

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

int main(){
    int x = 5;
    int y = 3;

    cout << x + y + 1 << endl;

    return 0;
}
```

4.33

4.33 (Lados de un triángulo recto) Escriba un programa que lea tres enteros distintos de cero, y que determine e imprima si podrían ser los lados de un triángulo recto.

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

int main(){
    int a, b, c;

    cout << "Ingrese los tres lados: ";
    cin >> a >> b >> c;

    if (a*a + b*b == c*c ||
        a*a + c*c == b*b ||
        b*b + c*c == a*a)
        cout << "Es un triangulo rectangulo." << endl;
    else
        cout << "No es un triangulo rectangulo." << endl;

    return 0;
}
```

4.35

4.35 (Inicializadores de listas de C++) Escriba instrucciones que usen la inicialización de listas de C++ para realizar cada una de las siguientes tareas:

- Inicializar la variable `unsigned int contadorEstudiantes` con 0.
- Inicializar la variable `double saldoInicial` con 1000.0
- Inicializar un objeto de la clase `Cuenta` que proporcione un constructor que reciba un `unsigned int`, dos `string` y un `double` para inicializar los miembros de datos `numeroCuenta`, `primerNombre`, `apellido` y `saldo`.

- `Unsigned contadorEstudiante{0};`
- `Double saldoInicial{1000.0}`
- `Cuenta cuenta{1234, "Juan", "Perez", 500.0}`

Link GitHub:

https://github.com/abner000r/Programacion_1_UMG.git