#include <iostream>

using namespace std;

int main() {

const int costoFijoDiario = 800;

const int costoVariablePorSombrero = 20;

const int costoPorSombreroObjetivo = 25;

int diasPorMes;

cout << "Ingrese la cantidad de días en el mes: ";

cin >> diasPorMes;

int costoFijoMensual = costoFijoDiario \* diasPorMes;

int sombrerosMinimos = (costoFijoMensual / (costoPorSombreroObjetivo - costoVariablePorSombrero)) + 1;

cout << "La cantidad mínima de sombreros que debe producir la fábrica mensualmente es: "

<< sombrerosMinimos << endl;

return 0;

}

Puse los dias de los meses por que algunos no son costantes tienen 30 o 31 dias.

EJERCICIO 2

#include <iostream>

using namespace std;

bool esNumeroEspecial(int numero) {

if (numero > 0 && numero % 3 == 0) {

if (numero % 100 == 0) {

return (numero % 300 == 0);

}

return true;

}

return false;

}

int main() {

int numero;

cout << "Ingrese un número: ";

cin >> numero;

if (esNumeroEspecial(numero)) {

cout << numero << " es un número especial." << endl;

} else {

cout << numero << " no es un número especial." << endl;

}

return 0;

}

EJERCICIO 3

#include <iostream>

using namespace std;

int main() {

int secuencia[5] = {3, 8, 12, 16, 21};

int siguiente[5];

int incrementos[4] = {4,4,5,5}; // Patrón de incrementos

int contador = 0; // Índice para el patrón

// Encontrar los siguientes 5 números

for (int i = 0; i < 5; ++i) {

// Calcular el siguiente número usando el último de la secuencia y el incremento correspondiente

int nuevoNumero = secuencia[4] + incrementos[contador];

siguiente[i] = nuevoNumero; // Guardar el nuevo número

secuencia[4] = nuevoNumero; // Actualizar el último número de la secuencia

contador = (contador + 1) % 4; // Cambiar al siguiente incremento

}

// Mostrar los siguientes números

cout << "Los siguientes 5 números en la secuencia son: ";

for (int i = 0; i < 5; ++i) {

cout << siguiente[i] << " ";

}

cout << endl;

return 0;

}

EJERCICIO 4

using namespace std;

bool esPrimo(int num) {

if (num <= 1) return false;

for (int i = 2; i \* i <= num; i++) {

if (num % i == 0) return false;

}

return true;

}

int main() {

vector<int> ficha(20);

int puntos = 0;

cout << "Ingrese 20 números: ";

for (int i = 0; i < 20; i++) {

cin >> ficha[i];

if (esPrimo(ficha[i])) puntos += 4;

if (ficha[i] % 2 == 0) puntos += 1;

if (ficha[i] % 7 == 0) puntos += 2;

}

cout << (puntos > 70 ? "¡Es una ficha ganadora!" : "No es una ficha ganadora.") << endl;

return 0;

}

EJERCICIO 5

#include <iostream>

using namespace std;

bool esNumeroEspejo(int numero) {

int original = numero;

int espejo = 0;

while (numero > 0) {

espejo = espejo \* 10 + (numero % 10);

numero /= 10;

}

return original == espejo;

}

int main() {

int ficha;

cout << "Ingresa el número de tu ficha (6 o 7 dígitos): ";

cin >> ficha;

if ((ficha >= 100000 && ficha <= 9999999) && (ficha < 1000000 || ficha >= 100000)) {

if (esNumeroEspejo(ficha)) {

cout << "Te tocó ir a las clases presenciales." << endl;

} else {

cout << "Te quedas en casita, ¡pero a estudiar!" << endl;

}

} else {

cout << "El número ingresado no es válido. Debe tener 6 o 7 dígitos." << endl;

}

return 0;

}

EJERCICIO 6

n = 9

while n > 0:

if n % 2 == 0:

print("Par")

else:

print("Impar")

n – = 3

La salida sera impar, par, impar y termina en 0 ya no podiendo imprimir mas.

EJRCICIO 7

#include <iostream>

using namespace std;

int main() {

int edad;

cout << "Introduce la edad del estudiante (entre 10 y 17): ";

cin >> edad;

if (edad < 10 || edad > 17) {

cout << "Edad no válida. Debe estar entre 10 y 17." << endl;

return 1;

}

if (edad % 5 == 0 && edad % 2 != 0) {

cout << "El estudiante pertenece al Grupo 1 (múltiplo de 5 pero no de 2)." << endl;

} else if (edad % 4 == 0) {

cout << "El estudiante pertenece al Grupo 2 (múltiplo de 4)." << endl;

} else if (edad % 3 == 0 && edad % 4 != 0 && edad % 5 != 0 && edad % 2 != 0) {

cout << "El estudiante pertenece al Grupo 3 (múltiplo de 3 pero no de 4, 5, ni 2)." << endl;

} else {

cout << "El estudiante pertenece al Grupo 4 (el resto)." << endl;

}

return 0;

}

EJERCICIO 8

if numero <= 1:

print("No")

else:

for i in range(2, int(numero\*\*0.5) + 1):

if numero % i == 0:

print("No")

else:

print("Sí")

en if el comienza a verificar menor o igual a 1 y si no es as imprime “no” ya que los números sin mayores que 1

y en el bucle for desde hasta la raíz cuadrado del numero con divisor mayor que su raíz cuadrada

el código verifica si el numero es divisible por i con modulo se imprime “no” indicando que no es primo y imprime si , “si” el numero es divisible por i.

EJERCICIO 9

#include <iostream>

using namespace std;

int main() {

int puntos;

double salarioAnual, bonificacion = 0;

cout << "Ingrese los puntos acumulados por el empleado: ";

cin >> puntos;

cout << "Ingrese el salario anual del empleado: ";

cin >> salarioAnual;

if (puntos < 50) {

bonificacion = 0;

} else if (puntos >= 50 && puntos <= 99) {

bonificacion = salarioAnual \* 0.05;

} else if (puntos >= 100) {

bonificacion = (salarioAnual \* 0.10) + 500;

}

cout << "La bonificación del empleado es: " << bonificacion << " Bs" << endl;

return 0;

}

EJERCICIO 10

#include <iostream>

using namespace std;

int main() {

int notas[10];

int suma = 0, promedio, notaAlta, notaBaja;

cout << "Ingrese las notas de 10 estudiantes:" << endl;

for (int i = 0; i < 10; i++) {

cout << "Nota del estudiante " << i + 1 << ": ";

cin >> notas[i];

suma += notas[i];

}

cout << "Notas ingresadas:" << endl;

for (int i = 0; i < 10; i++) {

cout << "Estudiante " << i + 1 << ": " << notas[i] << endl;

}

promedio = suma / 10;

cout << "El promedio de las notas es: " << promedio << endl;

notaAlta = notas[0];

notaBaja = notas[0];

for (int i = 1; i < 10; i++) {

if (notas[i] > notaAlta) {

notaAlta = notas[i];

}

if (notas[i] < notaBaja) {

notaBaja = notas[i];

}

}

cout << "La nota más alta es: " << notaAlta << endl;

cout << "La nota más baja es: " << notaBaja << endl;

return 0;

}

EJERCICIO 11

#include <iostream>

using namespace std;

int main() {

int costoBicicleta = 1200;

int ahorrado = 250;

int ahorroSemanal = 25;

int semanas = 0;

while (ahorrado < costoBicicleta) {

ahorrado += ahorroSemanal;

semanas++;

}

cout << "María necesita " << semanas << " semanas para ahorrar lo suficiente para comprar la bicicleta." << endl;

return 0;

}

EJERCICIO 12

#include <iostream>

using namespace std;

int main() {

double deuda = 3500;

double cuota = 0;

int numCuotas = 0;

cuota = 500;

deuda -= cuota;

numCuotas++;

cout << "Cuota " << numCuotas << ": " << cuota << " Bs" << endl;

while (deuda > 100) {

cuota = deuda \* 0.15;

deuda -= cuota;

numCuotas++;

cout << "Cuota " << numCuotas << ": " << cuota << " Bs" << endl;

}

if (deuda > 0 && deuda <= 100) {

cuota = deuda;

numCuotas++;

cout << "Cuota " << numCuotas << ": " << cuota << " Bs" << endl;

}

cout << "Número total de cuotas: " << numCuotas << endl;

return 0;

}

EJERCICIO 13

#include <iostream>

int main() {

int horaAlemania = 8;

int minutosAlemania = 0;

int diferenciaHoraria = 6;

int horaBolivia = horaAlemania - diferenciaHoraria;

if (horaBolivia < 0) {

horaBolivia += 24;

}

std::cout << "La hora boliviana para enviar el documento es: "

<< horaBolivia << ":" << (minutosAlemania < 10 ? "0" : "") << minutosAlemania

<< " (24 horas)" << std::endl;

return 0;

}

EJERCICIO 14

#include <iostream>

#include <string>

using namespace std;

string cifrarTexto(const string& texto) {

string textoCifrado = "";

for (char c : texto) {

if (c >= 'A' && c <= 'Z') {

textoCifrado += (c - 'A' + 1 < 10 ? "0" : "") + to\_string(c - 'A' + 1);

} else if (c >= 'a' && c <= 'z') {

textoCifrado += (c - 'a' + 1 < 10 ? "0" : "") + to\_string(c - 'a' + 1);

} else if (c == ' ') {

textoCifrado += "00";

}

}

return textoCifrado;

}

int main() {

string texto;

cout << "Ingrese el texto a cifrar: ";

getline(cin, texto);

string textoCifrado = cifrarTexto(texto);

cout << "Texto cifrado: " << textoCifrado << endl;

return 0;

}

EJERCICIO 15

#include <iostream>

#include <string>

using namespace std;

int main() {

const int METROS\_SALA = 100;

const int ANCHO\_SALA = 60;

const int AREA\_MESA = 2;

const int MESA\_CAPACIDAD = 3;

int totalMesas = (METROS\_SALA \* ANCHO\_SALA) / AREA\_MESA;

int totalParticipantes = totalMesas \* MESA\_CAPACIDAD;

string nombre;

int edad;

int participantesInscritos = 0;

while (participantesInscritos < totalParticipantes) {

cout << "Ingrese el nombre del participante: ";

getline(cin, nombre);

cout << "Ingrese la edad de " << nombre << ": ";

cin >> edad;

cin.ignore();

if (edad > 15 && edad < 30) {

if (participantesInscritos < totalParticipantes) {

cout << "¡Felicidades! " << nombre << ", eres parte del evento." << endl;

participantesInscritos++;

} else {

cout << "Lo siento, " << nombre << ", el evento está lleno." << endl;

}

} else {

cout << "Lo siento, " << nombre << ", tu edad no está en el rango definido." << endl;

}

}

cout << "El evento ha alcanzado su capacidad máxima de " << totalParticipantes << " participantes." << endl;

return 0;

}

EJERCICIO 16

#include <iostream>

using namespace std;

int main() {

const int NUM\_DADOS = 6;

int dados[NUM\_DADOS];

int suma = 0;

for (int i = 0; i < NUM\_DADOS; i++) {

cout << "Ingrese el valor del dado " << (i + 1) << " (1-6): ";

cin >> dados[i];

while (dados[i] < 1 || dados[i] > 6) {

cout << "Valor inválido. Ingrese el valor del dado " << (i + 1) << " (1-6): ";

cin >> dados[i];

}

suma += dados[i];

}

string resultado;

if (suma <= 8) {

resultado = "baja";

} else if (suma > 8 && suma <= 22) {

resultado = "mediana";

} else if (suma > 22 && suma <= 35) {

resultado = "alta";

} else if (suma == 36) {

resultado = "perfecta";

}

cout << "La suma de los dados es: " << suma << endl;

cout << "¡La jugada es " << resultado << "!" << endl;

return 0;

}

EJERCICIO 17

#include <iostream>

using namespace std;

int main() {

const double YENES = 285100;

const double DOLAR\_A\_YENES = 142.55;

const double DOLAR\_A\_BOLIVIANOS = 6.85;

const double FACTOR\_CAMBIO = 1.5;

double dolares = YENES / DOLAR\_A\_YENES;

double bolivianosOficial = dolares \* DOLAR\_A\_BOLIVIANOS;

double bolivianosReal = bolivianosOficial \* FACTOR\_CAMBIO;

cout << "El equivalente en dólares es: " << dolares << " $US" << endl;

cout << "El equivalente en bolivianos (cambio oficial) es: " << bolivianosOficial << " Bs" << endl;

cout << "El equivalente en bolivianos (cambio fuera de los bancos) es: " << bolivianosReal << " Bs" << endl;

if (dolares >= 1500) {

cout << "No requieres más fondos." << endl;

} else {

cout << "Debes conseguir más fondos." << endl;

}

return 0;

}

EJERCICIO 18

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

const double PRESTAMO = 14000;

const int MESES = 24;

double montoLaptop = PRESTAMO \* 0.50;

double montoMuebles = PRESTAMO \* 0.10;

double montoPublicidad = PRESTAMO \* 0.30;

double montoAplicaciones = PRESTAMO \* 0.10;

double interesLaptop = montoLaptop \* 0.05;

double interesMuebles = montoMuebles \* 0.07;

double interesPublicidad = montoPublicidad \* 0.10;

double interesAplicaciones = montoAplicaciones \* 0.12;

double totalAPagar = montoLaptop + interesLaptop +

montoMuebles + interesMuebles +

montoPublicidad + interesPublicidad +

montoAplicaciones + interesAplicaciones;

double pagoMensual = totalAPagar / MESES;

double totalInvertido = montoLaptop + montoMuebles + montoPublicidad + montoAplicaciones;

double sobrante = PRESTAMO - totalInvertido;

cout << fixed << setprecision(2);

cout << "Monto invertido en Laptop: " << montoLaptop << " Bolivianos" << endl;

cout << "Monto invertido en Muebles: " << montoMuebles << " Bolivianos" << endl;

cout << "Monto invertido en Publicidad: " << montoPublicidad << " Bolivianos" << endl;

cout << "Monto invertido en Aplicaciones: " << montoAplicaciones << " Bolivianos" << endl;

cout << "Cada mes, durante dos años debes pagar: " << pagoMensual << " Bolivianos" << endl;

if (sobrante > 0) {

double interesSobrante = sobrante \* 0.15;

cout << "Revisa tus inversiones, tienes " << sobrante << " Bolivianos con el interés más alto." << endl;

} else {

cout << "No tienes dinero sobrante." << endl;

}

return 0;

}

EJERCICIO 19

#include <iostream>

using namespace std;

int main() {

int tablasLargasDisponibles = 26;

int tablasCortasDisponibles = 33;

int ganchosPequeñosDisponibles = 200;

int ganchosGrandesDisponibles = 20;

int tornillosDisponibles = 510;

const int ESTANTES\_PEDIDOS = 15;

int tablasLargasRequeridas = 4 \* ESTANTES\_PEDIDOS;

int tablasCortasRequeridas = 6 \* ESTANTES\_PEDIDOS;

int ganchosPequeñosRequeridos = 12 \* ESTANTES\_PEDIDOS;

int ganchosGrandesRequeridos = 2 \* ESTANTES\_PEDIDOS;

int tornillosRequeridos = 14 \* ESTANTES\_PEDIDOS;

bool puedeCubrirPedido = (tablasLargasDisponibles >= tablasLargasRequeridas) &&

(tablasCortasDisponibles >= tablasCortasRequeridas) &&

(ganchosPequeñosDisponibles >= ganchosPequeñosRequeridos) &&

(ganchosGrandesDisponibles >= ganchosGrandesRequeridos) &&

(tornillosDisponibles >= tornillosRequeridos);

if (puedeCubrirPedido) {

cout << "Es posible realizar los 15 estantes pedidos." << endl;

int tablasLargasSobrantes = tablasLargasDisponibles - tablasLargasRequeridas;

int tablasCortasSobrantes = tablasCortasDisponibles - tablasCortasRequeridas;

int ganchosPequeñosSobrantes = ganchosPequeñosDisponibles - ganchosPequeñosRequeridos;

int ganchosGrandesSobrantes = ganchosGrandesDisponibles - ganchosGrandesRequeridos;

int tornillosSobrantes = tornillosDisponibles - tornillosRequeridos;

int estantesConSobrante = min(tablasLargasSobrantes / 4,

min(tablasCortasSobrantes / 6,

min(ganchosPequeñosSobrantes / 12,

min(ganchosGrandesSobrantes / 2,

tornillosSobrantes / 14))));

cout << "Con el material sobrante puedes construir " << estantesConSobrante << " estantes." << endl;

} else {

int faltantesTablasLargas = max(0, tablasLargasRequeridas - tablasLargasDisponibles);

int faltantesTablasCortas = max(0, tablasCortasRequeridas - tablasCortasDisponibles);

int faltantesGanchosPequeños = max(0, ganchosPequeñosRequeridos - ganchosPequeñosDisponibles);

int faltantesGanchosGrandes = max(0, ganchosGrandesRequeridos - ganchosGrandesDisponibles);

int faltantesTornillos = max(0, tornillosRequeridos - tornillosDisponibles);

cout << "No es posible cubrir el pedido porque faltan: "

<< faltantesTablasLargas << " tablas largas, "

<< faltantesTablasCortas << " tablas cortas, "

<< faltantesGanchosPequeños << " ganchos pequeños, "

<< faltantesGanchosGrandes << " ganchos grandes, "

<< faltantesTornillos << " tornillos." << endl;

}

return 0;

}

EJERCICIO 20

#include <iostream>

using namespace std;

int main() {

int tablasLargasDisponibles, tablasCortasDisponibles, ganchosPequeñosDisponibles;

int ganchosGrandesDisponibles, tornillosDisponibles, estantesPedidos;

cout << "Ingrese la cantidad de tablas largas disponibles: ";

cin >> tablasLargasDisponibles;

cout << "Ingrese la cantidad de tablas cortas disponibles: ";

cin >> tablasCortasDisponibles;

cout << "Ingrese la cantidad de ganchos pequeños disponibles: ";

cin >> ganchosPequeñosDisponibles;

cout << "Ingrese la cantidad de ganchos grandes disponibles: ";

cin >> ganchosGrandesDisponibles;

cout << "Ingrese la cantidad de tornillos disponibles: ";

cin >> tornillosDisponibles;

cout << "Ingrese la cantidad de estantes a construir: ";

cin >> estantesPedidos;

int tablasLargasRequeridas = 4 \* estantesPedidos;

int tablasCortasRequeridas = 6 \* estantesPedidos;

int ganchosPequeñosRequeridos = 12 \* estantesPedidos;

int ganchosGrandesRequeridos = 2 \* estantesPedidos;

int tornillosRequeridos = 14 \* estantesPedidos;

bool puedeCubrirPedido = (tablasLargasDisponibles >= tablasLargasRequeridas) &&

(tablasCortasDisponibles >= tablasCortasRequeridas) &&

(ganchosPequeñosDisponibles >= ganchosPequeñosRequeridos) &&

(ganchosGrandesDisponibles >= ganchosGrandesRequeridos) &&

(tornillosDisponibles >= tornillosRequeridos);

if (puedeCubrirPedido) {

cout << "Es posible realizar los " << estantesPedidos << " estantes pedidos." << endl;

int tablasLargasSobrantes = tablasLargasDisponibles - tablasLargasRequeridas;

int tablasCortasSobrantes = tablasCortasDisponibles - tablasCortasRequeridas;

int ganchosPequeñosSobrantes = ganchosPequeñosDisponibles - ganchosPequeñosRequeridos;

int ganchosGrandesSobrantes = ganchosGrandesDisponibles - ganchosGrandesRequeridos;

int tornillosSobrantes = tornillosDisponibles - tornillosRequeridos;

int estantesConSobrante = min(tablasLargasSobrantes / 4,

min(tablasCortasSobrantes / 6,

min(ganchosPequeñosSobrantes / 12,

min(ganchosGrandesSobrantes / 2,

tornillosSobrantes / 14))));

cout << "Con el material sobrante puedes construir " << estantesConSobrante << " estantes." << endl;

} else {

int faltantesTablasLargas = max(0, tablasLargasRequeridas - tablasLargasDisponibles);

int faltantesTablasCortas = max(0, tablasCortasRequeridas - tablasCortasDisponibles);

int faltantesGanchosPequeños = max(0, ganchosPequeñosRequeridos - ganchosPequeñosDisponibles);

int faltantesGanchosGrandes = max(0, ganchosGrandesRequeridos - ganchosGrandesDisponibles);

int faltantesTornillos = max(0, tornillosRequeridos - tornillosDisponibles);

cout << "No es posible cubrir el pedido porque faltan: "

<< faltantesTablasLargas << " tablas largas, "

<< faltantesTablasCortas << " tablas cortas, "

<< faltantesGanchosPequeños << " ganchos pequeños, "

<< faltantesGanchosGrandes << " ganchos grandes, "

<< faltantesTornillos << " tornillos." << endl;

}

return 0;

}

El algoritmo permite que las cantidades de cad item se ingrese en almacen asi como cantidad de pedido y se elimina los valores contantesy se recibira datos de entrada.

EJERCICIO 21

#include <iostream>

#include <vector>

using namespace std;

double calcularTarifa(double peso) {

if (peso <= 20) return 3.5;

else if (peso <= 50) return 4.5;

else if (peso <= 100) return 5.8;

else if (peso <= 200) return 7.0;

else if (peso <= 350) return 8.0;

else if (peso <= 500) return 9.0;

else if (peso <= 1000) return 12.0;

else if (peso <= 2000) return 17.0;

else if (peso <= 3000) return 21.0;

else return -1; // Peso fuera de rango

}

int main() {

int numPaquetes;

cout << "Ingrese el número de paquetes a enviar: ";

cin >> numPaquetes;

vector<double> pesos(numPaquetes);

double pesoTotal = 0.0;

double costoIndividual = 0.0;

for (int i = 0; i < numPaquetes; ++i) {

cout << "Ingrese el peso del paquete " << (i + 1) << " en gramos: ";

cin >> pesos[i];

costoIndividual += calcularTarifa(pesos[i]);

pesoTotal += pesos[i];

}

double costoTotal = calcularTarifa(pesoTotal);

cout << "\nResumen de tarifas:\n";

cout << "Costo total si se envían paquetes individualmente: " << costoIndividual << " Bs\n";

if (costoTotal != -1) {

cout << "Costo total si se envía todo junto: " << costoTotal << " Bs\n";

} else {

cout << "El peso total está fuera del rango para un envío convencional. Se considera como 'especial'.\n";

}

if (costoTotal == -1 && costoIndividual == -1) {

cout << "Todos los paquetes están fuera de rango. No se pueden enviar.\n";

} else if (costoTotal != -1 && (costoTotal < costoIndividual)) {

cout << "Es más conveniente enviar todos los paquetes juntos.\n";

} else {

cout << "Es más conveniente enviar los paquetes individualmente.\n";

}

return 0;

}

EJERCICIO 22

#include <iostream>

#include <cmath>

using namespace std;

int main() {

int cantidadMonedas = 6;

int diametroInicial;

cout << "Ingrese el diámetro de la primera moneda (entre 20 y 60 mm): ";

cin >> diametroInicial;

if (diametroInicial < 20 || diametroInicial > 60) {

cout << "El diámetro inicial debe estar entre 20 y 60 mm." << endl;

return 0;

}

int diametros[6];

diametros[0] = diametroInicial;

bool esPosible = true;

for (int i = 1; i < cantidadMonedas; ++i) {

double nuevoDiametro = diametros[i - 1] \* 1.3;

nuevoDiametro = round(nuevoDiametro);

if (nuevoDiametro < 20 || nuevoDiametro > 60) {

esPosible = false;

break;

}

diametros[i] = nuevoDiametro;

}

if (esPosible) {

cout << "Los diámetros de las 6 monedas son:" << endl;

for (int i = 0; i < cantidadMonedas; ++i) {

cout << "Moneda " << (i + 1) << ": " << diametros[i] << " mm" << endl;

}

} else {

cout << "No es posible crear las 6 monedas. Se pueden crear un total de " << (cantidadMonedas - 1) << " monedas." << endl;

}

return 0;

}

EJERCICIO 23

#include <iostream>

using namespace std;

int main() {

const int cantidadMonedas = 6;

const int diametroMinimo = 20;

const int diametroMaximo = 60;

int diametros[cantidadMonedas];

diametros[0] = diametroMinimo;

diametros[cantidadMonedas - 1] = diametroMaximo;

int incremento = (diametroMaximo - diametroMinimo) / (cantidadMonedas - 1);

for (int i = 1; i < cantidadMonedas - 1; ++i) {

diametros[i] = diametros[i - 1] + incremento;

}

cout << "Los diámetros de las 6 monedas son:" << endl;

for (int i = 0; i < cantidadMonedas; ++i) {

cout << "Moneda " << (i + 1) << ": " << diametros[i] << " mm" << endl;

}

cout << "Constante de incremento: " << incremento << " mm" << endl;

return 0;

}

EJERCICIO 24

for i in range(1, 10):

for j in range(1, 2):

i+=3

print(f"({i}, {j})", end=" ")

print()

1. Identificación de las variables:

variable del primer bucle i que toma valores del 1 al 9.

variable del segundo bucle j que siempre toma el valor de 1, ya que su rango es de 1 a 2 sin incluir el 2.

Las estrucuturas de control son

for: dos bucles for, uno anidado dentro del otro.

2. Explicación de las estructuras de control en este código:

El primer bucle for itera i desde 1 hasta 9. Por cada iteración de i, se ejecuta el segundo bucle for.

El segundo bucle for itera j desde 1 hasta 1, lo que significa que siempre se ejecutará una vez por cada valor de i.

Dentro del segundo bucle, la línea i += 3 incrementa el valor de i en 3. Esto significa que, aunque i está controlado por el primer bucle, se modificará durante la ejecución de este bucle, lo que afecta las iteraciones subsecuentes.

3. Impresión que generará este código:

La impresión resultante de este código será:

(4, 1) (8, 1)

Esto se debe a que i comienza en 1, se incrementa a 4 en la primera iteración (1 + 3), y luego, en la segunda iteración, se incrementa a 8 (4 + 3). El valor de j siempre será 1.

4. Impresión si se elimina la tercera línea de código:

Si se elimina la línea i += 3, la impresión será:

(1, 1) (2, 1) (3, 1) (4, 1) (5, 1) (6, 1) (7, 1) (8, 1) (9, 1)

En este caso, i se mantendrá constante en el rango del primer bucle, incrementándose automáticamente por el bucle, y se imprimirá junto con j que sigue siendo 1.