

Tarea 1 - Ejercicios básicos de C++

Esta tarea consiste en solucionar varios ejercicios de programación utilizando el lenguaje C++ y el ambiente de desarrollo Code::Blocks. Hay dos grupos de ejercicios, los primeros son ejercicios de programación básicos y los siguientes son ejercicios en los que se necesita utilizar arreglos o matrices para su solución.

Los ejercicios básicos se presentan a modo de práctica, no debe entregarse su solución, solamente la solución del ejercicio básico 12. Y de los ejercicios de arreglos y matrices el estudiante debe elegir únicamente dos ejercicios y presentar su solución.

¿Qué debe entregar?

Un archivo comprimido (ZIP) que contenga tres proyectos completos de Code::Blocks:

1. Ejercicio básico 12.
2. Ejercicio de arreglos y matrices elegido.
3. Ejercicio de arreglos y matrices elegido.

El archivo debe entregarse en el TEC-Digital mediante el módulo de evaluaciones.

Ejercicios básicos

1. Escriba un programa de C++ que pida al usuario que escriba dos números, obtenga los números del usuario e imprima la suma, producto, diferencia y cociente de los números.
2. Escriba un programa que lea el radio de un círculo como un número entero y que imprima su diámetro, circunferencia y área. Use el valor constante 3.14159 para π .
3. Escriba un programa que lea tres valores enteros y determine e imprima si los valores podrían ser los lados de un triángulo recto.
4. Escriba un programa que diga si un entero ingresado por el usuario es un palíndromo, es decir, un número que se lee igual para adelante que para atrás.
5. Escriba un programa de C++ que lea una serie de valores enteros, los ubique en un arreglo y determine si existe un par de elementos cuyo producto sea par.
6. Escriba un programa que reciba 20 números como entrada, cada uno de los cuales debe estar entre 10 y 100. A medida que se lee cada número, debe validar el rango y almacenarlo. Si el número ya se encuentra en el arreglo, no debe almacenarlo. Después de leer los valores, imprimirlos en pantalla, sólo los que no se repitieron. Utilice el arreglo más pequeño posible para resolver este problema.
7. Escriba programa de C++ que tome un arreglo con todos los números de 1 a 52 y los revuelva en orden aleatorio. Utilice la función `rand`, que retorna un entero pseudo-aleatorio cada vez que se llama. El resultado debe variar cada vez que se ejecuta la función.
8. Escriba un programa de C++ imprima todas las posibles cadenas de caracteres formadas por los caracteres 'a', 'b', 'c', 'd', 'e' y 'f' utilizados únicamente una vez.

9. El máximo común divisor, o MCD, de dos números enteros positivos n y m es el número más grande (j) tal que n y m son ambos múltiplos de j . Euclides propuso un algoritmo simple para calcular $\text{MCD}(n, m)$, donde $n > m$. La principal idea del algoritmo es efectuar repetidamente operaciones de residuo hasta llegar a cero. El último número diferente de cero en la secuencia obtenida es el MCD de n y m . Por ejemplo, para $n=80844$ y $m=25320$, la secuencia es la siguiente:

$$\begin{array}{rcl} 80844 \% 25320 & = & 4884 \\ 25320 \% 4884 & = & 900 \\ 4884 \% 900 & = & 384 \\ 900 \% 384 & = & 132 \\ 384 \% 132 & = & 120 \\ 132 \% 120 & = & 12 \\ 120 \% 12 & = & 0 \end{array}$$


De esta forma, el MCD de 80844 y 25320 es 12. Escriba un programa de C++ que lea dos números enteros y calcule su máximo común divisor.

10. Un castigo común a para los estudiantes de nivel escolar es escribir la misma oración muchas veces. Escriba un programa de C++ que escriba la siguiente oración cien veces: "Siempre debo utilizar diseño orientado a objetos". El programa debe numerar cada una de las oraciones y debe cometer ocho errores "accidentales" diferentes en varios puntos del listado, de forma que parezca que fue escrito por un humano.
11. Escriba un programa de C++ que reciba como entrada un día de la semana (de domingo a sábado) en un string y un entero de cuatro dígitos como año, y como resultado imprima el calendario de ese año. Cada mes debe tener el nombre del mes centrado sobre las fechas y una línea con los días de la semana (de domingo a sábado). Cada semana debe imprimirse en una línea separada. Tome en cuenta los años bisiestos.
12. El problema del cumpleaños, o paradoja del cumpleaños, dice que si hay al menos 23 personas reunidas, existe una probabilidad del 50% de que al menos dos personas cumplan años el mismo día. Esta propiedad no es una paradoja, pero mucha gente la encuentra sorprendente. Escriba un programa de C++ que pruebe este problema mediante una serie de experimentos con fechas de cumpleaños generadas aleatoriamente. Debe probar el problema con cantidad de fechas $n=5, 10, 15, 20, \dots, 100$. Ejecute al menos 10 experimentos por cada valor de n y escriba en la salida cuántos dieron como resultado fechas repetidas.

Ejercicios de arreglos y matrices

1. **Operaciones con matrices.** Escriba un programa que lea las dimensiones para crear una matriz. Estas dimensiones ($m \times n$) deben ser números pares para poder realizar una “reducción” del tamaño de la matriz a la mitad. El programa debe crear la matriz dinámicamente en memoria con valores aleatorios entre 0 y 100. Luego debe crear una nueva matriz con dimensiones $n/2 \times m/2$. El valor de cada celda de la nueva matriz es un promedio de un grupo de celdas de 2×2 de la matriz original. El programa debe imprimir en pantalla la matriz original y el resultado.

66	13	47	36	89	68
0	73	16	74	60	12
28	29	52	33	95	97
17	90	16	39	74	44



38	43.25	57.25
41	35	77.5

2. **Representación de polinomios.** Un polinomio cualquiera, por ejemplo $2x^5 + 3x^3 + 5x + 4$, puede representarse en un arreglo almacenando únicamente los coeficientes de los términos {4, 5, 0, 3, 0, 2}. Escriba un programa que lea los coeficientes de dos polinomios de grado 5, los presente en pantalla y luego imprima el resultado de la multiplicación y la suma de ambos polinomios. Tome en cuenta que luego de la multiplicación el grado del polinomio es mayor, por lo que necesita un arreglo de diferentes dimensiones para almacenar sus coeficientes.

Polinomio 1: $2x^5 + 3x^3 + 5x + 4$

{4, 5, 0, 3, 0, 2}

Polinomio 2: $4x^4 + x^3 + 5x^2 + 2$

{2, 0, 5, 1, 4, 0}

Suma: $2x^5 + 4x^4 + 4x^3 + 5x^2 + 5x + 6$

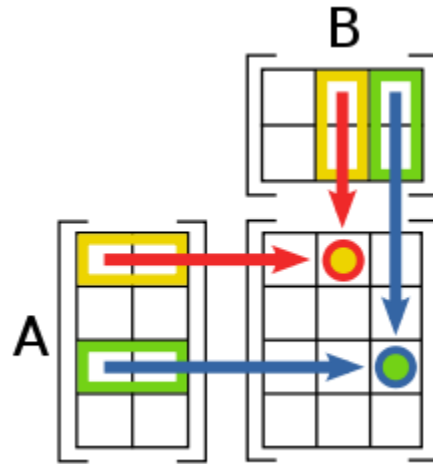
{6, 5, 5, 4, 4, 2}

Mult.: $8x^9 + 2x^8 + 22x^7 + 3x^6 + 39x^5 + 21x^4 + 35x^3 + 20x^2 + 10x + 8$

{8, 10, 20, 35, 21, 39, 3, 22, 2, 8}

3. **Estadísticas de frecuencias.** Escriba un programa de C++ que lea una cantidad N de números. Primero, el programa debe solicitar la cantidad de números que van a ser procesados. Seguidamente debe leer del teclado los números y almacenarlos en un arreglo creado dinámicamente del tamaño que indicó el usuario inicialmente. Luego de leer todos los números, para cada uno de ellos deben calcularse e imprimirse las siguientes estadísticas:
 - a. Frecuencia absoluta de cada número: número de veces que aparece el número en la serie.
 - b. Frecuencia absoluta acumulada: número de veces que aparecen los valores menores o iguales a cada valor en la serie.
 - c. Frecuencia relativa acumulada: es la frecuencia absoluta acumulada dividida entre N.
4. **Multiplicación de matrices.** Escriba un programa que calcule la multiplicación de dos matrices A y B de números enteros y retorne el resultado en una nueva matriz C. Deben leerse las dimensiones de las matrices al inicio (recuerde que para poder multiplicarlas sus dimensiones

deben ser $m \times n$ y $n \times p$, respectivamente). Los valores de las matrices pueden calcularse aleatoriamente o solicitarse al usuario.

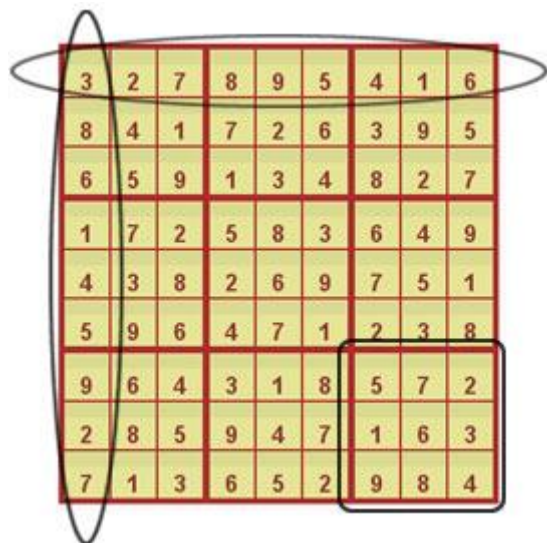


5. **Área de un octágono.** La siguiente fórmula permite calcular el área de cualquier triángulo (no necesariamente rectángulo) a partir de sus tres vértices (x_1, y_1) , (x_2, y_2) y (x_3, y_3) .

$$A = \left| \frac{x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)}{2} \right|$$

Escriba un programa que lea desde el teclado las coordenadas de ocho puntos que en secuencia definen un octágono irregular pero convexo, y posteriormente imprima el área de dicho octágono, calculada mediante la aplicación de la fórmula anterior. Investigue cómo determinar si los puntos del octágono son válidos y cómo calcular un punto central en la figura para aplicar la fórmula de área repetidas veces.

6. **Caballo de ajedrez.** Escriba un programa en C++ que presente un tablero de ajedrez vacío y pregunte al usuario una posición para ubicar un caballo. La idea del programa es que el caballo vaya moviéndose por el tablero sin repetir casillas ya visitadas y efectuando solamente movimientos válidos. Cada movimiento del caballo debe elegirse de manera aleatoria entre todos los movimientos posibles. El tablero debe representarse internamente mediante una matriz y debe dibujarse en pantalla cada uno de los movimientos realizados. Los movimientos deben continuar hasta que no hayan movidas válidas. Al finalizar el programa debe presentar la cantidad de celdas visitadas, sin visitar y el porcentaje de las casillas que fueron visitadas por el caballo.
7. **Sudoku.** Escriba un programa que lea una matriz de 9x9 e indique si los valores ingresados conforman un sudoku válido. Un sudoku es válido si los números del 1 al 9 no se repiten en la misma fila, en la misma columna o en la misma sub-área de 3x3.



3	2	7	8	9	5	4	1	6
8	4	1	7	2	6	3	9	5
6	5	9	1	3	4	8	2	7
1	7	2	5	8	3	6	4	9
4	3	8	2	6	9	7	5	1
5	9	6	4	7	1	2	3	8
9	6	4	3	1	8	5	7	2
2	8	5	9	4	7	1	6	3
7	1	3	6	5	2	9	8	4