

Universidad Nacional Autónoma de Honduras

Facultad de Ciencias

Guías de ejercicios

Programación I MM314/IC200

Escuela de matemáticas y ciencias de la computación

Departamento de matemática aplicada

✧ Guía cuarto foro

1. Una matriz de probabilidad es una matriz cuadrada que tiene dos propiedades:

- (a) Todos sus elementos son no negativos.
- (b) La suma de los elementos en cada renglón es 1.

Determine si una matriz dada es una matriz de probabilidad.

2. En un taller de m participantes se practicaron 5 pruebas. Los resultados de cada prueba se leerán en un arreglo N , donde cada n_{ij} es la nota del participante i en la prueba j ; Calcular:

- (a) La nota promedio de cada participante.
- (b) El mayor promedio obtenido y el correspondiente número de su participante.

3. Sea la matriz cuadrada $A_{n \times n}$, donde cada elemento a_{ij} se encuentra en la fila i y en la columna j . Calcule e imprima :

- (a) El producto de los elementos de la diagonal principal.
- (b) La suma de aquellos elementos para los cuales $i < j$.
- (c) El cuadrado de la suma de los elementos si $i > j$.

4. Haga un programa que haga las operaciones con matrices: suma, resta, producto escalar, multiplicación, opuesta y transpuesta.

5. Sea el arreglo $A_{m \times n}$. Lea m , n y los elementos de la matriz. Calcule el producto de los elementos que se encuentran en la intersección de la horizontal y la vertical con el elemento a_{ij} , donde un elemento k que se lee debe coincidir con a_{ij} ; este elemento no debe incluirse en el producto. Considerar sólo la primera coincidencia de k con a_{ij} .

6. Una compañía constructora tiene tres proyectos de construcción de casas : A, B y C. Por cada proyecto ha agrupado los costos en cuatro tipos, a saber :

- (a) Costos por diseño,
- (b) Costos por materiales de construcción,
- (c) Costos por suelos y salarios y
- (d) Costos por administración.

Los costos se almacenan en un arreglo bidimensional $X_{3 \times 4}$, donde i en x_{ij} representa el proyecto y j , en el mismo elemento, denota el tipo de costo. Escriba un programa que calcule:

- (a) el costo total por diseño de los tres proyectos,
- (b) el costo total por materiales de construcción,
- (c) el costo total por sueldos y salarios,
- (d) el costo total por administración,
- (e) el costo total por proyecto y
- (f) el costo total por los tres proyectos.

7. Dado un número entero n con $n \geq 2$, escribir la matriz identidad de dimensión $n \times n$.

8. Dada una matriz $A_{m \times n}$ y un vector X_n ; Se desea generar e imprimir un nuevo vector Y_m , que se forma realizando las siguientes operaciones:

$$\begin{aligned} Y[1] &= A[1][0] \times X[0] + A[1][1] \times X[1] + \dots + A[1][n] \times X[n] \\ Y[2] &= A[2][0] \times X[0] + A[2][1] \times X[1] + \dots + A[2][n] \times X[n] \\ &\vdots \\ Y[m] &= A[m][0] \times X[0] + A[m][1] \times X[1] + \dots + A[m][n] \times X[n] \end{aligned}$$

9. Sea el arreglo $A_{m \times n}$. Lea m , n y los elementos de la matriz. Calcule e imprima el valor máximo de cada fila y el valor mínimo de cada columna.

10. Una compañía tiene cuatro vendedores (1 a 4) que venden cinco productos (1 a 5). Una vez por día cada vendedor entrega una hoja por cada tipo de producto vendido, la cual contiene: El número del vendedor, el número del producto y el importe de las ventas de dicho producto en el día. Por lo tanto, cada vendedor entrega entre 0 y 5 hojas de ventas por día. Suponga que están a la mano las hojas de información de las ventas del último mes. Escriba un programa que tome como entrada esta información y resuma las ventas totales por vendedor y producto. Todos los totales deben almacenarse en el arreglo bidimensional ventas. Tras procesar la información del último mes, imprima el resultado en formato de tabla - que cada columna represente un vendedor y cada fila represente un producto. Totalice cada fila para obtener las ventas totales por producto; totalice cada columna para determinar las ventas totales por vendedor. La impresión tabular deberá incluir estos totales a la derecha de las filas y al final de las columnas.

11. El departamento de policía de la ciudad de Tegucigalpa ha acumulado información referente a las infracciones de los límites de velocidad durante un determinado periodo de tiempo. El departamento ha dividido la ciudad en cuatro cuadrantes y desea realizar una estadística de las infracciones a los límites de velocidad en cada uno de ellos. Para cada infracción se ha preparado una tarjeta que contiene la siguiente información:

- (a) número de placa del vehículo;
- (b) cuadrante en el que se produjo la infracción
- (c) límite de velocidad en milla por hora

Diseñe un programa para producir 2 informes que contienen:

- (a) el primero, una lista de la multa de velocidad recolectadas, donde la multa se calcula como la suma del costo de la corte (L 00) mas L 12.50 por cada mph que exceda la velocidad limite. Prepare una tabla con los siguientes resultados:

INFRACCIONES A LOS LÍMITES DE VELOCIDAD			
Placa del vehículo	velocidad registrada (MPH)	velocidad límite	Multa

- (b) el segundo, un análisis de las infracciones por cuadrante. Para cada uno de los 4 cuadrantes mencionados, debe darse el número de infracciones y la multa promedio.
12. Diez equipos de la liga Inter-barrial identificados con los números 1, 2, 3, ... , 10, participaron en un campeonato de fútbol en la modalidad todos contra todos. Los goles anotados en cada encuentro se registraron en el siguiente cuadro:

	1	2	3	4	...	10
1	0	4	2	1	...	
2	2	0	3	2	...	
3	0	2	0	1	...	
\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	
10						

Se puede ver por ejemplo que el equipo 1 marcó 4 al equipo 2, 2 goles al equipo 3, 1 gol al equipo 4 etc. El equipo 2 marcó 2 goles al equipo 1, 3 goles al equipo 3, etc.

Escriba un programa que:

- Simule el cuadro de goles generando de forma aleatoria un arreglo bidimensional
 - muestre el total de de goles recibidos por equipo
 - muestre el equipo que más goles anoto
 - determine el números juegos empatados en el torneo
13. Estudios científicos informan que del 26 al 5 de mayo se sentirá más calor de lo habitual debido a la cercanía del planeta tierra al sol. Conforme a la rotación de la misma, lo que provocara que los rayos caigan perpendicularmente en Honduras. El departamento de Meteorología de la UNAH desea hacer un análisis de las temperatura que se registraran esos días para lo que se medirán las temperatura a las 11:50 en los 18 departamentos (1-18). Mediante un arreglo bidimensional:
- Resuma las temperaturas promedios por día y por departamento.
 - Calcule el número de departamentos que registren una temperatura mayor de 35 grados centígrados
 - Determine los departamento que el día 28 de abril sobrepasaron la temperatura promedio de todos estos días
 - Calcule cuantos días el departamento # 10 registro una temperatura entre 36 a 40 grados centígrados.
 - Determine la mayor temperatura registrada durante ese período y escriba el o los departamentos y el o los días que alcanzaron esa temperatura.
14. Elabore un programa de consulta y reserva de aulas del edificio F1. Para ello debe:
- crear de forma aleatoria una matriz booleana $F1[10][2][4]$ (10 aulas, 2 líneas, 4 pisos); donde 0 indique aula vacía y 1 aula reservada. Las aulas están enumeradas con 3 dígitos, el dígito de la izquierda indica el piso (del 1 al 4), y los dos últimos dígitos el número del aula, las aulas del 1 al 10 están en la línea 1(izquierda) y las aulas del 11 al 20 en la línea 2(derecha), por ejemplo el aula 411 está ubicada en el cuarto piso en la línea 2 y es la primera aula y el aula 206 se encuentra en el segundo piso en la línea 1 y es la sexta aula.
 - el programa debe realizar las siguientes acciones:
 - Indicar si hay aulas disponibles o si todas las aulas están ocupadas.
 - Listar las aulas vacías de un determinado piso y/o línea

- Listar las aulas ocupadas de un determinado piso y/o línea.
 - Consulta de un aula en particular.
 - Reservar aulas disponibles.
 - Mostrar la matriz de un determinado piso o de los cuatro pisos.
15. Escriba un programa que trabaje con un arreglo de caracteres bidimensional de tamaño 8×8 que simule un tablero de ajedrez. El programa deberá inicialmente imprimir la tabla con formato de tablero de ajedrez, para esto puede colocar el símbolo numeral como los escaques negros y espacios en blanco para los escaques blancos. Luego aleatoriamente se colocará la letra *R* representante de la reina, a continuación el programa solicitará al usuario una posición a la cual desearía mover la reina siempre que el movimiento sea permitido y se visualizará en pantalla este movimiento.
 16. Escriba una función que reciba un arreglo bidimensional de tamaño 4×4 de forma que se interprete como una matriz, luego la función debe retornar el determinante de esta matriz.
 17. Escriba una función que calcule la matriz inversa de una matriz de tamaño 4×4 usando el determinante. Para lo anterior la función recibirá 2 arreglos bidimensionales de tamaño 4×4 .
 18. Implemente una función que determine los índices del elemento con valor mínimo en una tabla de números reales.
 19. Desarrolle la siguiente función:

```
void Buscar(vector <int> & X, int A[ ],int n)
```

Esta función debe buscar el arreglo A dentro del vector X. El argumento n representa el tamaño del arreglo A.

20. Desarrolle la siguiente función:

```
void Almacenar(vector <char> & X)
```

Dentro de la función se almacenarán en X varios caracteres desde el teclado con ayuda de un bloque de repetición. Termine la lectura cuando se almacene una secuencia de caracteres iguales a la palabra "final".

21. Simule el juego de tic tac toe con un arreglo bidimensional de tamaño 3×3 .
22. Use un arreglo bidimensional de tamaño 10×10 para almacenar los primeros 100 números primos y visualice esta tabla por pantalla con formato de tabla.
23. Escriba un programa que trabaje con un arreglo de caracteres bidimensional de tamaño 8×8 que simule un tablero de ajedrez. El programa deberá inicialmente imprimir la tabla con formato de tablero de ajedrez, para esto puede colocar el símbolo numeral como los escaques negros y espacios en blanco para los escaques blancos. Luego aleatoriamente se colocará la letra *R* representante de la reina, a continuación el programa solicitará al usuario una posición a la cual desearía mover la reina siempre que el movimiento sea permitido y se visualizará en pantalla este movimiento.
24. Construya la siguiente función:

```
void Funcion5(vector <int> & A,int j)
```

Esta función deberá eliminar el elemento del vector A que se encuentra en la posición j. Si el índice está fuera de rango, entonces escriba un mensaje que indique esta situación.

25. Construya la siguiente función:

void Funcion3(vector <int> & A,vector <int>& B, vector <int>& C)

Esta función debe almacenar todos lo elementos en común de A y B en C, con la única condición que no se repitan los elementos.

Considere el siguiente ejemplo:

A={1,1,2,2,3,4,4,5}

B={4,5,6,7,8,9,101}

Después de usar esta función, C quedaría de la siguiente forma:

C={4,5}

26. Desarrolle un programa en c++ que dado un arreglo de orden n, muestre en pantalla la suma de las posiciones con números primos.
27. Desarrolle un programa en c++ que dado un arreglo de orden n, muestre en pantalla la siguiente sucesión $A[0] + A[n - 1], A[1] + A[n - 2], \dots$
28. Desarrolle un programa en c++ que dado un arreglo de orden n, muestre en pantalla la siguiente sucesión $A[0] * A[n - 1], A[1] * A[n - 2], \dots$
29. Dado un arreglo de caracteres de orden n , desarrolle un programa que imprima en pantalla todas las posibles combinaciones de letras que se pueden formar con dicho arreglo.
30. Suponga que se quiere imprimir un cuadrado como el de la siguiente figura suponiendo que conocemos previamente el valor de N . Implemente una función en C++ que imprima tal cuadrado según sea el valor de N .

```

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
1 2 3 3 3 3 3 3 3 3 3 3 3 3 2 1
1 2 3 4 4 4 4 4 4 4 4 4 4 4 3 2 1
1 2 3 4 5 5 5 5 5 5 5 5 5 5 4 3 2 1
1 2 3 4 5 6 6 6 6 6 6 6 6 6 5 4 3 2 1
1 2 3 4 5 6 7 7 7 7 7 7 7 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 8 8 8 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 9 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 9 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 8 8 8 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 7 7 7 7 7 6 5 4 3 2 1
1 2 3 4 5 6 6 6 6 6 6 6 6 5 4 3 2 1
1 2 3 4 5 5 5 5 5 5 5 5 5 5 4 3 2 1
1 2 3 4 4 4 4 4 4 4 4 4 4 4 4 3 2 1
1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

```

$N = 10$

```

1 1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 2 1
1 2 3 3 3 3 2 1
1 2 3 4 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 4 3 2 1
1 2 3 3 3 3 2 1
1 2 2 2 2 2 2 1
1 1 1 1 1 1 1 1 1

```

$N = 5$

31. Construya la siguiente función:

int Mochila(vector <int> V, vector <int> P, int k)

Dentro de la función se deben ordenar los elementos de P en orden decreciente, a la vez se deben de reorganizar los elementos en V bajo este orden. Luego se debe retornar el índice i tal que la suma de los elementos en P desde el índice cero hasta el índice i sea máxima y menor o igual que k. Considere el siguiente ejemplo:

$$V=[23,34,45,22,78] \quad P=[57,12,89,34,56] \quad k=150$$

Después de usar la función obtendríamos:

$$V=[45,23,78,22,34] \quad P=[89,57,56,34,12]$$

y la función retornaría $i=1$, puesto que $P[0]+P[1]=89+57=146 \leq k=150$.

32. Construya la siguiente función:

int Mochila1(vector <double> & V, vector <double> & P, int k)

Dentro de la función se deben ordenar los elementos de P y V en orden decreciente en función de los cocientes entre los elementos de P y V. Luego se debe retornar el índice i tal que la suma de los elementos en P desde el índice cero hasta el índice i sea máxima y menor o igual que k. Considere el siguiente ejemplo:

$$P=[60,34,45,25,56] \quad V=[10,2,15,5,7] \quad k=150 \quad \text{Cocientes}=[60/10,34/2,45/15,25/5,56/7]=[6,17,3,5,8]$$

Después de usar la función obtendríamos:

$$P=[34,56,60,25,45] \quad V=[2,7,10,5,15] \quad \text{Cocientes}=[17,8,6,5,3]$$

y la función retornaría $i=2$, puesto que $P[0]+P[1]+P[2]=34+56+60=150 \leq k=150$.

33. Escriba una función que reciba un arreglo bidimensional de tamaño 4×4 de forma que se interprete como una matriz, luego la función debe retornar el determinante de esta matriz.
34. Escriba una función que calcule la matriz inversa de una matriz de tamaño 4×4 usando el determinante. Para lo anterior la función recibirá 2 arreglos bidimensionales de tamaño 4×4 .
35. Suponga que una mano de cartas es representada por un arreglo bidimensional de tamaño 2×5 , de manera que la primera fila representa el número de la carta(1,2,3,...,13) y la segunda fila represente el palo($1 \equiv \clubsuit$, $2 \equiv \diamondsuit$, $3 \equiv \heartsuit$, $4 \equiv \spadesuit$). Implementar una función que imprima todas las combinaciones de dobles, triples, cuádruples, quintuples y escaleras de esta mano.
36. Construya la siguiente función:

void Funcion6(vector <int> & A,int x)

Esta función deberá eliminar todos los elemento del vector A que son iguales a x. Si el valor no está en el vector, simplemente el vector no se verá modificado.

37. En un determinado casino hay un juego de azar que consiste en tirar dos dados y sumar los números de las caras superiores. Observe que las sumas de las caras pueden ser de un mínimo de 2 y de un máximo de 12. Cada número que surge de estas sumas en un lanzamiento tiene asociado una Ganancia o pérdida. La fórmula que se utiliza para determinar la ganancia o pérdida dado que se obtuvo el número “k” en un lanzamiento es: $Premio = 50 * (|7 - k| - 2)$ Lempiras, donde $k = 2, 3, 4, \dots, 12$. Observe que el premio puede ser positivo (cuando $k = 2, 3, 4, 10, 11, 12$) o puede ser negativo (cuando $k = 6, 7, 8$); cuando el premio es positivo significa que el jugador gana la cantidad que dice la fórmula y cuando es negativo significa que el jugador pierde la cantidad que dice

la fórmula. Un jugador empedernido a juegos de azar quiere realizar un experimento para determinar si el juego es justo o a la larga el jugador siempre lleva las de perder. El jugador se propone jugar durante 30 días consecutivos realizando 10 lanzamientos por día y ver si al final acumuló ganancia o pérdida. El jugador quiere hacer una simulación de dicho experimento antes de arriesgarse a perder dinero jugando de verdad. Elabore un programa que simule el experimento; para ello haga lo siguiente:

- Construir un arreglo de enteros bidimensional de tamaño 30×10 con nombre L que corresponde a los resultados obtenidos de los 10 lanzamientos de los dados por día, durante los 30 días. Para simular los resultados de los lanzamientos, se generan dos números aleatorios que oscilan entre 1 y 6, por tanto los elementos de L es la suma de estos dos números aleatorios.
- Imprimir el arreglo L de forma tabulada por filas y columnas.
- A partir de la matriz L determine la ganancia o pérdida acumulada al final de los 30 días. Usar la fórmula "Premio" para calcular los premios por cada resultado de los lanzamientos registrados en L y acumularlos. Fórmula: $Premio = 50 * (|7 - k| - 2)$ Lempiras, donde " k " es el resultado del lanzamiento.
- Construya un arreglo unidimensional con nombre FR de tamaño 11 cuyos elementos son las frecuencias de cada uno de los resultados registrados en L , es decir, el primer elemento de FR es el número de veces que se obtuvo el resultado 2 hasta el, el segundo elemento es el número de veces que se obtuvo el resultado 3, y así sucesivamente.
- Imprimir de forma tabulada los resultados (2, 3, 4, ..., 12) y su respectiva frecuencia.
- Ordenar e imprimir el arreglo FR de menor a mayor.

38. Suponga que se tiene un arreglo bidimensional de 4×4 , este arreglo contiene números naturales entre 1 y 16 (incluyendo al 1 al 16) sin repeticiones.

- Desarrolle una función booleana que determine si en la matriz anterior al sumar los elementos en cada fila, resulta que las cuatro sumas dan resultados iguales.
- Hacer lo mismo que en el inciso anterior ahora con las columnas.
- Elabore una función booleana que determine si en la matriz anterior al sumar los elementos en cada diagonal, resulta que las dos sumas son iguales.
- Desarrollar una función que combine las tres funciones anteriores para verificar si las diez sumas son iguales.

39. David y Licien deciden jugar *Suerte con el dado*. "Suerte con el dado" es un juego de dados; para jugarlo se necesita una tabla de dos filas por cuatro columnas y dos dados. Abajo se ve el ejemplo de la tabla.





David				
Licien				

Las reglas del juego son las siguientes:

- Inicialmente cada jugador llena sus 4 casillas con números de dos dígitos, donde estos dígitos se encuentren el rango de 1 a 6.
- En cada ronda ambos jugadores tiran los dos dados, si la combinación de estos dos números es uno de los números de la lista entonces este lo tacha.
- No hay un número definido de rondas. Gana el jugador que tache todos sus números.

Por ejemplo, imagine que Licien y David llenan inicialmente la tabla como aparece a continuación:

David	16	23	43	45
Licien	14	12	52	61

Luego suponga que David lanza  y  mientras que Licien tira  y , entonces David tacharía el 16 y Licien no tacharía nada puesto que su tiro no coincide con ninguno de sus dígitos en la tabla.

Después de esto la tabla se vería así.

David	16	23	43	45
Licien	14	12	52	61

Desarrolle un programa en C++ que simule este

juego.

- (a) Implemente la función *Inicial*

```
void Inicial(int T[ ][4])
```

El arreglo bidimensional *T* representa la tabla de jugadores, el objetivo de la función es llenar la tabla usando *rand*, de acuerdo a las reglas.

- (b) Implemente la función *Ronda*

```
void Ronda(int T[ ][4])
```

El objetivo de la función es simular los dos tiros de los jugadores usando la función *rand*. Luego de ello para representar que un número ha sido tachado en lugar del número se escribirá el valor -1.

- (c) Dentro del programa principal simule el juego con ayuda de las funciones anteriores y al final indique quien es el ganador y el número de rondas jugadas.

40. Imagine un juego en el que un apostador lanza tres dados, luego de esto debe revisar todas las posibles combinaciones de su tiro. Si la combinación observada suma un múltiplo de 3, debe añadir el resultado a su puntuación(inicialmente empieza con 0 puntos) ó de lo contrario no acumula nada. Para ilustrar lo anterior imagine que un jugador tira $\{\square, \boxplus, \boxplus\}$, entonces si observamos todas las posibles combinaciones $\{\square\}$, $\{\boxplus\}$, $\{\boxtimes\}$, $\{\square, \boxplus\}$, $\{\square, \boxtimes\}$, $\{\boxplus, \boxplus\}$, $\{\square, \boxplus, \boxplus\}$ notamos que las únicas combinaciones que suman un múltiplo de tres son $\{\boxplus, \boxplus\}$ y $\{\square, \boxplus\}$, de esta forma en este tiro el jugador acumularía $9+6=15$ puntos.

El apostador tiene oportunidad de lanzar los tres dados 5 veces, además gana si acumula más de 100 puntos en estas 5 oportunidades. El apostador tiene miedo de perder su dinero en este juego, por tal motivo para ayudar al apostador escriba un programa en C++ que simule este juego.

- (a) Desarrolle una función con el siguiente prototipo:

```
void TiroTriple(int &x, int &y, int &z)
```

la cual almacenará en *x*, *y* y *z*, con ayuda de la función *rand*, los tres dados tirados.

- (b) Desarrolle una función con el siguiente prototipo:

```
int Combinaciones(int x, int y, int z)
```

la cual recibe como entrada los tres dados tirados y retornará el puntaje obtenido por este tiro.

- (c) Con ayuda de las funciones anteriores en el programa principal simule los 5 oportunidades del jugador y al final mande un mensaje que indique si ganó o perdió.

41. Suponga que Jennifer, Martha y Luisa se disponen a participar en un juego de dados.

Para jugar se necesita disponer de una tabla de 4 columnas por tres filas. Abajo se ve un ejemplo de tal tabla:

Jennifer				
Martha				
Luisa				

El juego consta de 4 partidas. Durante cada partida cada jugador lanza un dado y anota el resultado en la fila correspondiente y la columna que este desee(siempre que no este ocupada). Al final del juego se tomará en cuenta el número de cuatro dígitos que se lea en cada fila; el número más alto será el ganador.

Un ejemplo de como podría haber terminado este juego se ve en la siguiente tabla:

Jennifer	6	1	2	3
Martha	5	4	5	3
Luisa	6	3	4	1

En este ejemplo habría ganado Luisa pues es el número más alto(6,341).

Desarrolle un programa en C++ que simule este juego.

- (a) Implemente la función *Genera*, la cual tiene como prototipo

```
void Genera(int T[ ][4])
```

Esta función recibe como entrada una tabla de 3 filas por 4 columnas la cual simula la tabla del juego. La función debe simular una partida almacenando los tiros de dados en la tabla con ayuda de la función *rand*.

- (b) Con ayuda de la función anterior simule todo el juego y al final imprima la tabla después de la cuarta partida indicando mediante un mensaje quien es el ganador del juego.

42. Desarrolle un programa que llene aleatoriamente con valores de 0 a 10000 un arreglo bidimensional T de 1200×1200 elementos. Se define la *bola* centrada en el elemento $T[i][j]$ como el conjunto de entradas $T[k][l]$ tales que $|k - i| < 3$, $|l - j| < 3$. Defina la intensidad de la bola centrada en $T[i][j]$ como la media aritmética de los elementos que pertenecen a esta bola. Encuentre el elemento tal que la bola centrada en él tenga intensidad máxima. Para esto debe imprimir los índices donde esto sucede.

43. Construya la siguiente función:

`void DivisoresPrimos(int n, vector <int> & DP)`

Esta función tiene como objetivo guardar en el vector DP todos los divisores diferentes y primos de n . En la función principal pida al usuario que ingrese el valor de n y construya el vector DP que contiene los divisores primos de n con ayuda de la función anterior. Finalmente imprima este vector.

44. Construya la siguiente función:

`void SoloPrimos(vector <int> & X, vector <int> & SP)`

El objetivo de esta función es almacenar en SP solo los elementos de X que sean números primos. En la función principal pida al usuario que llene con números enteros positivos un arreglo X y construye con ayuda de la función que se describió antes el arreglo SP que contiene exclusivamente los primos de X. Imprima a SP.

45. Construya las siguientes funciones:

`void Divisores(int n, vector <int> & V)`

Esta función debe almacenar en V todos los divisores (tanto positivos como negativos) de n .

`void DivisoresCruz(vector <double> & A, vector <double> & B, vector <double> & C)`

Esta función deberá guardar en C todas las posibles divisiones entre los elementos de A y los elementos de B, sin agregar elementos repetidos.

Use las funciones anteriores para simular el resultado conocido en álgebra como el teorema de las raíces racionales.

46. Construya la siguiente función:

`void Partir(vector <int> & X, vector <int> & A, vector <int> & B)`

Esta función ordenará en forma creciente los elementos de X. Luego en A almacenará los números pares contenidos en X sin agregar repetidos. En B de manera similar se agregarán los impares de X, nuevamente sin repetir elementos. Construya un ejemplo en el programa principal y muestre la salida de los resultados de la función.

47. Crear un programa que empiece a leer números enteros positivos y los vaya almacenando en un vector que está vacío inicialmente. Termine la lectura cuando el usuario ingrese un número negativo y no almacene este último valor. Finalmente imprima la moda y la mediana de este conjunto de datos.
48. Crear un programa que empiece a leer números enteros positivos y los vaya almacenando en un vector que está vacío antes de empezar la lectura. Termine de hacer la lectura cuando el usuario haya ingresado una secuencia de 4 números consecutivos de Fibonacci. Finalmente imprima el vector resultante.

49. Suponga que una persona tiene una lista de gastos. Esta persona quiere saber si hay una secuencia de gastos que sumen una cantidad X . Cree un programa donde se define inicialmente un vector vacío y luego se irán almacenando los gastos. Finalice la lectura de los gastos cuando alguna sucesión de estos sumen la cantidad X ó cuando el usuario ingrese un número negativo.
Considere la posibilidad de que esta persona pueda equivocarse desde el teclado al ingresar los gastos, dele la posibilidad de corregir este error.
50. Construya un programa en C++ que contenga la función `VerificadorDeOrden`, que reciba como parámetro un arreglo de enteros y devuelva un valor booleano que indique si los elementos del arreglo están ordenados o no.
51. Construya un programa en C++ que dado un arreglo de n enteros, determine si cada elemento es el doble del anterior o no, y en caso contrario, retorne la posición del primer elemento inválido.
52. Construya un programa en C++ que permita llenar un arreglo de orden n , en donde cada elemento debe almacenar un caracter (que debe ser ingresado desde teclado). Luego, se deberá procesar el arreglo de tal forma que despliegue por pantalla los valores almacenados pero con corrimiento como se muestra a continuación:

```

FRANCISCO
RANCISCOF
ANCISCOFR
NCISCOFRA
CISCOFRAN
ISCOFRANC
SCOFRANCI
COFRANCIS
OFRANCISC
FRANCISCO

```

53. En un arreglo de n elementos, se almacenan valores numéricos enteros (positivos y/o negativos). Para agregar un valor al arreglo, éste se debe solicitar por teclado y luego buscar la posición en donde debería estar almacenado para mantener el orden dentro del arreglo. Sin embargo, si el valor ya existe, no se debe permitir almacenarlo; en tal caso, se debe mostrar el mensaje “YA EXISTE”. Cada vez que se ingrese un nuevo valor, se deberá mostrar por pantalla todos los valores almacenados (esto servirá para determinar si el almacenamiento se está realizando correctamente)
54. Generar un arreglo de n números aleatorios entre 0 y 100. A partir de él se pide:
- Obtener el valor máximo y la posición en que se encuentra.
 - Obtener el valor mínimo y la posición en que se encuentra.
 - Dividir el arreglo en dos: A y B. El A debe contener los elementos de índices pares y el B los elementos de índices impares.
 - Mostrar en pantalla los arreglos A y B
55. Desarrolle un programa en c++ que dado arreglo de enteros de orden n , muestre en pantalla los divisores de los elementos en las posiciones pares.
56. Construya la siguiente función:

```
void Funcion(vector<vector<int>> &A)
```

Esta función debe ordenar los elementos de A en forma creciente, considerandolo como una tabla en sentido de izquierda a derecha y de arriba abajo.

Por ejemplo, si A es la siguiente tabla:

$$A = \begin{pmatrix} 3004 & -3 & 23 \\ -11 & 33 & 101 \\ 1 & 2345 & 545 \end{pmatrix}$$

Entonces dentro de la función se debe modificar A de la siguiente forma:

$$A = \begin{pmatrix} -11 & -3 & 1 \\ 23 & 33 & 101 \\ 545 & 2345 & 3004 \end{pmatrix}$$

57. Construya la siguiente función:

`void EliminarRepetidos(vector <int> X)`

Esta función debe eliminar todos los elementos repetidos dentro del vector X . Por ejemplo si el vector X contiene los elementos $\{1, 1, 2, 4, 5, 5, 6, 7, 8, 1, 1, 1001, 2, 2, 2\}$, entonces al ejecutar esta función X quedaría conformado con los siguientes elementos $\{1, 2, 4, 5, 6, 7, 8, 1001\}$.

58. Desarrolle la siguiente función:

`void Funcion1(vector <int> V)`

Esta función de expandir el vector en la siguiente forma: El elemento en la posición $V[0]$ se copiará dentro del vector $V[1]$ veces. Luego el elemento en la posición $V[2]$ se copiará dentro del vector $V[3]$ veces, y así sucesivamente. Considere el siguiente ejemplo: Suponga que el argumento se ingresa el siguiente vector:

$$V = [1, 2, 3, 4, 10, 3, 4, 5]$$

Entonces dentro de la función el vector V se convertiría en:

$$V = [1, 1, 3, 3, 3, 3, 10, 10, 10, 4, 4, 4, 4, 4]$$

Asuma para este ejercicio que el vector V tiene una cantidad par de elementos.

59. Construya la siguiente función:

`void Funcion2(vector <int> & A, vector <int> & B, vector <int> & C)`

Esta función debe almacenar todos los elementos de A y B en C , con la única condición que no se repitan los elementos.

Considere el siguiente ejemplo:

$$\begin{aligned} A &= \{1, 1, 2, 2, 3, 4, 5\} \\ B &= \{4, 5, 6, 7, 8, 9, 101\} \end{aligned}$$

Después de usar esta función, C quedaría de la siguiente forma:

$$C = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 101\}$$

60. Construya la siguiente función:

void Funcion4(vector <int> & A,vector <int>& B, vector <int>& C)

Esta función debe almacenar en C todos los elementos de A que no están en B sin repetir elementos. Considere el siguiente ejemplo:

$$A=\{1,3,4,1,1,5,5,6,8\} \quad B=\{5,8,15,1\}$$

Entonces los elementos que se deben almacenar en C son:

$$C=\{3,4,6,8\}$$

61. La sucesión de Fibonacci se define como.

$$f_0=0$$

$$f_1=1$$

$$f_n=f_{n-1} + f_{n-2}$$

desarrolle un programa que almacene un arreglo con los primeros m elementos de la sucesión de Fibonacci.

62. Desarrolle un programa que dado un arreglo de orden n , llene sus posiciones con el factorial de cada posición,
63. Desarrolle un programa que dado un arreglo de orden n genere m números aleatorios entre 0 y n , el programa deberá de contabilizar las veces que se repite cada valor entre 0 y n .
64. Desarrolle un programa que dado un arreglo de orden n , imprima en pantalla el número que más se repite, en caso de que exista más de uno con la misma frecuencia deberá de imprimir el promedio de ellos.
65. Desarrolle un programa en c++ que dado un arreglo $h[n]$ que contiene únicamente valores positivos entre 1 y 100, muestre en pantalla la siguiente matriz:

$$\begin{array}{cccccc} 1 & 0 & 0 & 0 & \dots & 0 \\ h[0] & 2 * (h[0] + h[1]) & h[1] & 0 & \dots & 0 \\ 0 & h[1] & 2 * (h[1] + h[2]) & h[2] & \dots & 0 \\ 0 & 0 & h[2] & 2 * (h[2] + h[3]) & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & h[n-1] & \ddots & 1 \end{array}$$

66. Desarrolle un programa que dado dos arreglos, ambos de dimension n , calcule la suma del producto de ambos arreglos, ejemplo: $A[0] * B[0] + A[1] * B[1] + \dots + A[n-1] * B[n-1]$
67. Desarrolle un programa que dado dos arreglos, ambos de dimension n , calcule la suma del producto de las posiciones pares de ambos arreglos, ejemplo: $A[0] * B[0] + A[2] * B[2] + \dots$
68. Desarrolle un programa que dado dos arreglos, ambos de dimension n , calcule la suma del producto de las posiciones impares de ambos arreglos, ejemplo: $A[1] * B[1] + A[3] * B[3] + \dots + A[n-1] * B[n-1]$
69. Desarrolle un programa que dado dos arreglos, ambos de dimension n , calcule la suma del producto de ambos arreglos, considerando la siguiente forma: $A[0] * B[1] + A[1] * B[2] + \dots + A[n-1] * B[0]$
70. Desarrolle un programa que dado un arreglo de n filas y m , el arreglo deberá de llenarse de forma aleatoria con valores entre 10 y 99 y visualizado en pantalla en forma de matriz, luego el usuario ingresara un entero y se deberá presentar en pantalla la suma de todas las posiciones (Fila, Columna) donde se encuentra el entero ingresado.

71. Desarrolle un programa que dado un arreglo de n filas y m , el arreglo deberá de llenarse de forma aleatoria con valores entre 10 y 99 y visualizado en pantalla en forma de matriz, luego el usuario ingresara un entero y se deberá presentar en pantalla la suma de todas las filas donde se encuentra el entero ingresado.
72. Desarrolle un programa que dado un arreglo de n filas y m , el arreglo deberá de llenarse de forma aleatoria con valores entre 10 y 99 y visualizado en pantalla en forma de matriz, luego el usuario ingresara un entero y se deberá presentar en pantalla la suma de todas las columnas donde se encuentra el entero ingresado.