

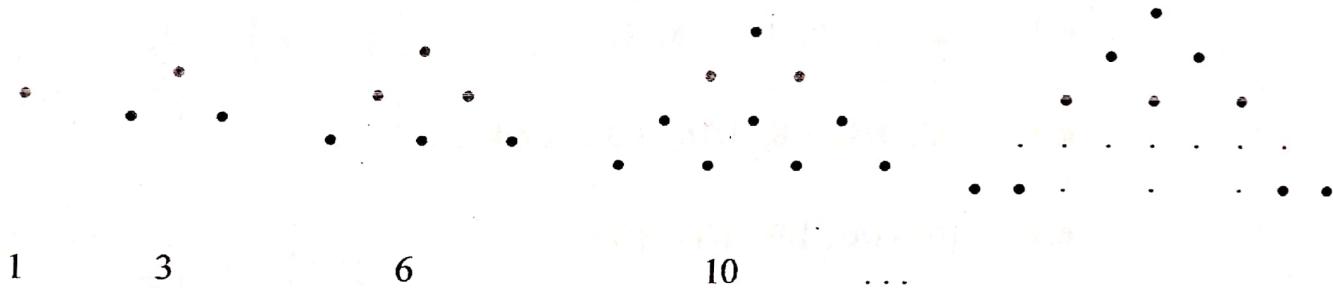
## GUÍA DE EJERCICIOS No. 2

### PARTE I: FUNCIONES

1. Para cada inciso, se pide:

Resumen del Análisis, Seudocódigo, Prueba de Escritorio y codificación en C++

- a. Los cuadrados perfectos son los números 1, 4, 9, 16, . . . Escriba los primeros n números, donde n es un número natural mayor que 0.
- b. Considere la secuencia de los números triangulares: 1, 3, 6, 10, . . . , cuyo nombre refleja su ley de formación :



Escriba los primeros n números triangulares, donde n es un número natural mayor que 0. Además, y en particular, haga que el penúltimo número se acompañe del mensaje “Este es el penúltimo número triangular”.

- c. Escriba un programa que presente en la pantalla el siguiente triángulo formado con números dígitos:

```
    1
   232
  34543
 4567654
 567898765
 67890109876
 7890123210987
 890123454321098
 90123456765432109
 0123456789876543210
```

- d. Dibuje en la pantalla la siguiente figura, compuesta por líneas de  $2, 4, 8, 16, \dots, 2^n$  respectivamente:

A decorative separator at the top of the page. It features a central vertical column of asterisks (\*). Above this central column are three rows of asterisks: one row above the central column, one row below it, and one row below that. Below the central column of asterisks is a horizontal line of asterisks. At the bottom of the page is a horizontal line of dots (.) followed by a horizontal line of asterisks (\*).

- e. Escribir un programa que imprima las siguientes series de números:

- e.1.  $0, -3, 6, -9, 12, -15, 18, \dots$ ,

e.2.  $0, 1, 1, 2, 3, 5, 8, 13, 21, \dots$

e.3.  $-2, 6, -10, 14, -18, 22, \dots$

e.4.  $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \dots$

e.5.  $\frac{1}{3}, -\frac{1}{6}, \frac{1}{9}, -\frac{1}{12}, \frac{1}{15}, \dots$

- f. Escriba un programa que calcule la potencia de un número entero no negativo definiendo función potencia(n).
  - g. Escriba un programa que calcule la potencia de un número entero no negativo definiendo función recursiva rpotencia(n).
  - h. Escriba un programa que calcule el factorial de un número entero no negativo definiendo una función factorial(n).
  - i. Escriba un programa que calcule el factorial de un número entero no negativo definiendo una función recursiva rfactorial(n).

j. Use las funciones del inciso g e i en un programa que dado x, n calcule e imprima:

$$j.1. \quad 1 + \frac{x^2}{1!} + \frac{x^3}{2!} + \frac{x^n}{3!} + \dots + \frac{x^n}{n!}$$

$$j.2. \quad \sum_{i=1}^n \left( \frac{1}{2}i - \frac{7}{2} \right) \frac{x^{2(i+1)}}{i!}$$

$$j.3. \quad \sum_{i=0}^n \frac{(-1)^i x^{2(i+1)}}{(i+1)! (2i)!} \quad \text{Seno}(x)$$

$$j.4. \quad \sum_{i=0}^n \frac{(x-i)^2 + 2i}{i!}$$

$$j.5. \quad \sum_{i=0}^n \frac{(-1)^i x^{2i}}{(2i)!} \quad \text{Coseno}(x)$$

k. Escriba un programa que imprima 10 líneas de asteriscos a partir de un número Natural dado n, con  $1 \leq n \leq 70$ . El programa debe tener una función que recibe un numero x e imprima una linea de x asteriscos.

- I. Escriba un programa que lea un número binario y escriba su equivalente en base diez (vea ejemplo). El programa debe verificar que el número dado por el usuario sea binario, además usara las funciones siguientes:
- CONTAR: recibe el número dado por el usuario y retorna la cantidad de dígitos que tiene.
  - VALIDAR: recibe el número dado por el usuario y la cantidad de dígitos que tiene; verifica que el número sea binario retornando el valor booleano adecuado.
  - CONVERTIR: recibe uno de los dígitos del número y la posición que ocupa; retorna el valor relativo decimal.

Veamos dos ejemplos para convertir un número binario a decimal.

Ejemplo 1. Convertir 11101 a base 10

$$1x2^0 + 0x2^1 + 1x2^2 + 1x2^3 + 1x2^4 = 29_{10}$$

Ejemplo 2. Convertir 1000 a base 10

$$0x2^0 + 0x2^1 + 0x2^2 + 1x2^3 = 8_{10}$$

m. Hacer un programa que calcule la siguiente suma.

$$\text{Sumf} = E(0) + \frac{E^3(1)}{2!}(3) + \frac{E^5(2)}{3!}(4) + \frac{E^7(3)}{4!}(5) \dots$$

$$\text{Donde se define } E(X) = \sum_{N=0}^L \frac{1}{N!} X^{N+2}$$

- n. Se requiere de un programa que calcule el área de la superficie y el volumen de una esfera, dado su radio. El programa principal, lee el radio de la esfera y usa las funciones `Menu()`, `AreaEsfera(radio)` y `VolumenEsfera(radio)` para hacer los cálculos respectivos que serán impresas en el programa principal.
- o. Escriba un programa que acepte como entrada un conjunto de números enteros positivos y los pase, uno a la vez, a la función *primo*, que determina si el número que recibe es primo o no.(retorna un valor booleano) Al final el programa debe escribir la cantidad de números primos y la cantidad de números compuestos que se ingresaron.  
Nota: El 1 no es primo ni compuesto.
- p. Haga un programa que lea varias ecuaciones cuadráticas (una a la vez) de la forma  $a x^2 + b x + c = 0$  y escriba la concavidad, el eje de simetría, el vértice y intercepto con el eje y intercepto con el eje x y los intervalos de crecimiento y decrecimiento de cada una de ellas. Implemente funciones para cada elemento.
- q. Haga un programa que genere aleatoriamente 10 ecuaciones lineales (una a la vez) de la forma  $a x + b y + c = 0$  ( $a \neq 0$  y  $b \neq 0$ ) y solicite al usuario la pendiente, el intercepto en x y el intercepto en y. Al final el programa debe escribir el porcentaje de respuestas correcta. La respuesta es correcta si no falla en ninguno de los tres elementos que se piden

2. Reescriba los siguientes programas implementando las funciones que le indican.

a. Funciones: Perímetro y Área

Inicio

$\text{pi} \leftarrow 3.1416$

Para  $i \leftarrow 1, 5$

Lea  $r$

$\text{perimetro} \leftarrow 2(r)(\text{pi})$

$\text{area} \leftarrow \text{pi}(r)^2$

Escriba "El perímetro del círculo es: ", perímetro

Escriba "El área del círculo es: ", área

Final

b. Funciones: Menú, Suma, Resta, Multiplica, Divide

Inicio

Repita

Lea  $x, y$

Escriba "1. Sumar"

Escriba "2. Restar"

Escriba "3. Multiplicar"

Escriba "4. Dividir"

Escriba "0. Terminar"

Lea opc

Selección opc

Caso 1:  $\text{resp} \leftarrow x + y$

Caso 2:  $\text{resp} \leftarrow x - y$

Caso 3:  $\text{resp} \leftarrow xy$

Caso 4: Si  $y \neq 0$

$\text{Resp} \leftarrow x/y$

Sino

Escriba "División No definida"

Por defecto: Si  $\text{opc} \neq 0$

Escriba "Opción no válida"

Si  $1 \leq \text{opc} \geq 4 \wedge y \neq 0$

Escriba resp

Hasta que  $\text{opc} = 0$

Final

c. Funciones: para f1, f2, f3

Inicio

lea Vi, Vf, paso  
escriba "x ----- f(x)"  
para x  $\leftarrow$  Vi(paso)Vf  
si x < -2  
    y  $\leftarrow$   $x^2 - 3x - 2$  //f1

sino si x < 2  
    y  $\leftarrow$   $3 - 2x$  //f2  
    sino  
        y  $\leftarrow$   $1 - x^2$  //f3  
escriba x, "-----", y

Final

3. Haga la prueba de escritorio de cada programa.

a. Datos 8, 5

inicio

Lea p  
cP (p)  
escriba "Numero:"  
lea n  
cR (n)

fin

cP (m)

inicio

para i  $\leftarrow$  1, m  
    si i < m escriba i, ","  
    si no escriba i

fin

cR ( m )

inicio

para j  $\leftarrow$  m,-1, 1  
    si j > 1 escriba j, ","  
    si no escriba j

fin

b.

Inicio

n  $\leftarrow$  4  
num  $\leftarrow$  0  
para i  $\leftarrow$  0, n  
    para k  $\leftarrow$  0, n  
        escriba num  
        si k  $\neq$  n  
            num = Siguiente (num, n)

Final

Siguiente ( nn, tope )

Inicio

si nn = tope

retorne 0

si no

retorne nn+1

Final

c. Datos 7, 4

Inicio

Escriba "Introduzca un 1er. entero : "

Lea x

Haga

Escriba "Introduzca un 2do. Entero (debe ser positivo) : "

Lea y

Mientras y  $\leq$  0

Escriba "El resultado es " , Misterio ( x, y )

Final

Misterio (a, b)

Inicio

si b = 1

retorne a

si no

retorne a + Misterio ( a, b - 1 )

Final

d. Datos: 5

```
Inicio
    suma ← 0;
    lea n
    para i ← 1,n
        num ← R(i)
        den ← F(i)
        suma ← suma + S(i) * C(num, den);
    escriba suma
Final
```

F(m)

```
Inicio
    Si m=1
        retorne m
    sino
        retorne m * F(m-1)
Final
```

C(m, n)

```
Inicio
    retorne m/n
Final
```

R(m)

```
Inicio
    retorne raiz cuadrada de m
Final
```

S(m)

```
Inicio
    retorne (-1)i+1
Final
```

e. Datos 5, 2, 1, 4, 3, 1, 7, 7, 0

Inicio

repita

Lea n1, n2

Escriba nCm (n1, n2)

Escriba "desea hacer otro calculo 1: si/ 0: no"

Lea r

Hasta que r = 0

Final

nCm (n, m)

Inicio

Si n = 0 v m = 0

retorne 1

si no

retorne nCm (n- 1, m- 1) + nCm (n- 1, m)

Final

f. Datos 3, 0, 0, 4, 6, 9

Inicio

haga

lea n1, n2

mientras n1 ≠ 0 ^ n2 ≠ 0

escriba "MCD(“, n1, ”, ”, n2 , “) = “, MCD (n1, n2)

Final

MCD (x, y)

Inicio

si y = 0

retorne x

si no

retorne MCD(y, x mod y)

Final

g. Datos: 4, 3, 2, 1, 0

Inicio  
repita  
    escriba "Cantidad de discos "  
    lea n  
    escriba "El numero de movimientos es ", Hanoi(n)  
    hasta que n = 0  
Final

Hanoi(n)  
Inicio  
    si n = 1  
        retorne 1  
    si no  
        retorne 2 \* Hanoi( n - 1 ) + 1  
Final

## PARTE II: ARREGLOS UNIDIMENSIONALES 0.

1. Para cada inciso, se pide Análisis, Resumen del análisis, Seudocódigo, Prueba de Escritorio y codificación en C++ (Implemente funciones en los programas si es posible).
  - a. Calcule el residuo de la división de  $P(x) / Q(x)$  donde el grado del polinomio P es mayor o igual a dos y el polinomio Q es de la forma  $x - r$  con  $r \neq 0$ .

Implemente una función que calcule el residuo mediante la división sintética y otra función que lo haga por medio del teorema del residuo.

- b. Calcule la suma de los primeros  $n$  elementos de  $a_n = a_{n-1}^2 - 2a_{n-2}$ , donde  $a_0 = 1.5$ ,  $a_1 = 1$  y  $n \geq 2$  (incluya en la cuenta los dos datos suministrados).

- c. Sean 1000 valores representados por  $a_1, a_2, a_3, \dots, a_{1000}$ . Calcule e imprima  $S_1, S_2, S_3, \dots, S_{100}$ , donde :

$$S_1 = \frac{a_1 + a_2 + \dots + a_{10}}{10},$$

$$S_2 = \frac{a_{11} + a_{12} + \dots + a_{20}}{10},$$

$$\dots$$
  
$$S_{100} = \frac{a_{991} + a_{992} + \dots + a_{1000}}{10}.$$

- d. Se leen las componentes de un vector n-dimensional. Determine si tal vector es un vector nulo y si no lo es, encuentre el número de elementos distintos de cero y la posición que éstos ocupan en el vector.
- e. Escriba un programa que genere de forma aleatoria un arreglo unidimensional de n elementos (se lee n), de tal modo que escriba el arreglo en dos formas:
- del primer al último elemento:  $a_1, a_2, \dots, a_n$
  - del último elemento al primero:  $a_n, a_{n-1}, \dots, a_1$
- f. Lea los valores  $x_i$  del vector X y calcule la expresión
- $$\frac{x_3 x_4 x_5}{2!} - \frac{x_5 x_6 x_7}{4!} + \frac{x_7 x_8 x_9}{6!} - \dots + \frac{x_{43} x_{44} x_{45}}{42!}$$
- g. Lea los valores  $x_i$  apropiados y calcule la expresión
- $$\frac{x_0 x_1 x_2}{1!} - \frac{x_2 x_3 x_4}{3!} + \frac{x_4 x_5 x_6}{5!} - \dots + \frac{x_{38} x_{39} x_{40}}{39!}$$
- h. Calcule  $\sum_{j=1}^m \frac{1}{m-1} (y_j - \text{med})^2$ , donde  $\text{med} = \frac{1}{m} \sum_{j=1}^m y_j$ . haga función med.
- i. Lea un vector de 20 números naturales entre 50 y 100, ambos inclusive. A medida que se lea cada número, almacenarlo solamente si no es repetido. Al final escriba el vector.
- j. Genere de forma aleatoria un arreglo unidimensional de cinco números entre 1 y 25, ambos inclusive. Ningún número del arreglo puede ser el doble de otro. Escriba el arreglo.
- k. Escriba un programa que calcule  $\sum_{i=1}^n 3x_i^2$ . Se leen n y  $x_i$ , donde  $n \geq 1$ .
- l. Escriba un programa que calcule e imprima  $\sum_{i=1}^n (x_i^2 - y_i^2)$ . Se leen n,  $x_i$  e  $y_i$ , donde  $n \geq 1$ .
- m. Sean las componentes de dos vectores:  $X_1, X_2, \dots, X_n$  y  $Z_1, Z_2, \dots, Z_n$  para  $n \geq 2$ . Intercambiar los contenidos de los elementos de la siguiente manera:

$X_1$  en  $Z_n$  y viceversa,  
 $X_2$ , en  $Z_{n-1}$  y viceversa, ...,  
 $X_{n-1}$  en  $Z_2$  y viceversa,  
 $X_n$  en  $Z_1$  y viceversa.

- n. Escriba un programa que lea un arreglo unidimensional de N números enteros, e implemente funciones para encontrar:
- \* el mayor elemento.
  - \*\* Frecuencia del mayor el mayor elemento.
  - \*\*\* posiciones en que se ubica el mayor elemento dentro del vector
- o. Escriba un programa que lea una arreglo unidimensional X de tamaño N y determine si el número n dado por el usuario se encuentra ó no en el arreglo; si se encuentra escribir las posiciones donde se ubica dentro del arreglo.
- p. Escriba un programa que lea un vector X de n números enteros positivos y a partir de él genere los arreglos Xpar y Ximpar los que contendrán los números pares e impares de X respectivamente.

2.- Realice la prueba de escritorio de los siguientes programas y codifíquelo en C++

a. Datos 3, 5, 2, 3, 2, 2, 3.

```

inicio
... lea m, n
... para i  $\leftarrow$  1, m
.... ci  $\leftarrow$  0
.... para j  $\leftarrow$  1, n
..... si i = 1 lea bj
..... si bj = i ci  $\leftarrow$  ci + 1
... para j  $\leftarrow$  1, m
.... escriba j, " ", cj
fin

```

b. Datos 5, 7, 6, 4, 8, 2

Inicio

```

lea n
para i  $\leftarrow$  1, n
    Lea a[ i ]
para i  $\leftarrow$  1, n
    Escriba a[ i ]
para pasada  $\leftarrow$  1, n-1
    para i  $\leftarrow$  1, n - 1
        si a[ i ] > a[ i + 1 ]
            temp  $\leftarrow$  a[ i ]
            a[ i ]  $\leftarrow$  a[ i + 1 ]
            a[ i + 1 ]  $\leftarrow$  temp
para i  $\leftarrow$  1, n
    escriba a[ i ]

```

Final

c. Datos: 3, 9, 5, 4, 0, 7, 1, 8, 6, 2

Inicio  
tamano  $\leftarrow$  10  
para i  $\leftarrow$  1, tamano  
    lea  $x_i$   
    resultado  $\leftarrow$  QueSeraEsto( $x[ ]$ , tamano)  
    escriba "El resultado es ", resultado  
Final

QueSeraEsto(  $y[ ]$ , tam )

Inicio  
    si tam = 1  
        retorne  $y_1$   
    si no  
        retorne  $y_{tam} +$  QueSeraEsto( $y[ ]$ , tam - 1);  
Final

d. Datos 5, 10, 4, 5, 1, 3, 1, 2, 3, 4, 2, 2

inicio  
    lea m, n  
    para i  $\leftarrow$  1, m  
         $c_i \leftarrow 0$   
        para j  $\leftarrow$  1, n  
            si  $b_j = i$   
                lea  $b_j$   
                si  $b_j = i$   
                     $c_i \leftarrow c_i + 1$   
            para j  $\leftarrow$  1, m  
                escriba "El ", j, "tiene una frecuencia de ",  $c_j$   
final

e.

Inicio

$n \leq 10$   
 $x[n] \in \{32, 27, 64, 18, 95, 14, 90, 70, 60, 37\}$   
 $PT(b[], n)$

Final

$PT(b[], m)$

Inicio

si  $m > 0$   
    escriba  $b[m], "$   
     $PT(b[], m-1)$

Sino cambio de linea

Final

f. D1: 4, 4, 3, 2, 1     D2: 3, 2, 1, 0 (son dos pruebas de escritorio)

Inicio

$p \leftarrow 1$   
 $cin >> n;$   
para  $i \leftarrow 1, n$   
    lea  $a[i]$   
 $i \leftarrow 0$   
mientras  $a[i] \neq 0$   
     $p = p * a[i]$   
     $i \leftarrow i + 1$

si  $i < n$   
    escriba "p = 0"

sino  
    escriba "p = ", p

Final

g. Datos 5, 4, 0, -0.5

Inicio

Lea grado  
Para  $i \leftarrow 0$ , grado  
     $P[i] \leftarrow \text{azar}(10)^*(i) ^ \wedge (\text{azar}(2))$

Lea V<sub>i</sub>, V<sub>f</sub>, paso

$Y[] \leftarrow \{0\}$

Para  $i \leftarrow V_i$  (paso) V<sub>f</sub>

    Para  $j \leftarrow 0$ , grado

$Y[i] \leftarrow i^j * P[j] + Y[i]$

    Escriba  $i, "-----", Y[i]$

Final

### PARTE III: ARREGLOS BIDIMENSIONALES

1.- Para cada caso, presente: Breve Análisis, Seudocódigo, Prueba de Escritorio y codificación en C++  
Implemente funciones en los programas.

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

- todos sus elementos son no negativos y
- la suma de los elementos en cada renglón es 1.

Determine si una matriz es una matriz de probabilidad.

- b. 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

- la nota promedio de cada participante y
- el mayor promedio obtenido y el correspondiente número de su participante.

- c. 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 :

- El producto de los elementos de la diagonal principal;
- La suma de aquellos elementos para los cuales  $i < j$
- El cuadrado de la suma de los elementos si  $i > j$ .

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

- e. 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 1era. coincidencia de  $k$  con  $a_{ij}$ .

- f. 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 :

- Costos por diseño,
- Costos por materiales de construcción,
- Costos por sueldos y salarios y
- Costos por administración.

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

- el costo total por diseño de los tres proyectos,
- el costo total por materiales de construcción,
- el costo total por sueldos y salarios,
- el costo total por administración,
- el costo total por proyecto y
- el costo total por los tres proyectos.

- g. Dado un número entero  $n$  con  $n \geq 2$ , escribir la matriz identidad de dimensión  $n \times n$ .
- h. 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:

$$Y[1] = A[1][0] * X[0] + A[1][1] * X[1] + \dots + A[1][n] * X[n]$$

$$Y[2] = A[2][0] * X[0] + A[2][1] * X[1] + \dots + A[2][n] * X[n]$$

$$\dots$$

$$Y[m] = A[m][0] * X[0] + A[m][1] * X[1] + \dots + A[m][n] * X[n]$$

- i. 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.

- j. 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.

- k. 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:

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

Diseñe un programa para producir 2 informes que contienen:

- 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 LIMITES DE VELOCIDAD			
Placa del vehículo	Velocidad registrada (MPH)	Velocidad limite	Multa

- 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.

- I. 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:

		Equipos						
		1	2	3	4	...	10	
Equipos	1	0	4	2	1	...		
	2	2	0	3	2	...		
		3	0	2	0	1	...	
		10						

Se puede ver por ejemplo que el equipo 1 marcó 4 goles 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 goles recibidos por equipo
- muestre el equipo que más goles anoto
- determine el número de juegos empatados en el torneo

- m. 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 temperaturas que se registraran esos días para lo que se medirán las temperaturas a las 11:50 en los 18 departamentos (1-18). Mediante un arreglo bidimensional:

- Resuma las temperaturas promedio por día y por departamento.
- Calcule el número de departamentos que registren una temperatura mayor de 35 grados centígrados
- Determine los departamentos que el día 28 de abril sobrepasaron la temperatura promedio de todos estos días
- Calcule cuantos días el departamento #10 registró 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.

2.- Realice la Prueba de Escritorio

a. Dato: 3

Inicio

lea n

para i  $\leftarrow$  1, n

para j  $\leftarrow$  1, n

si i  $\leq$  j

$a_{ij} \leftarrow 2i + j$

si no si i  $>$  j

$a_{ij} \leftarrow i * j$

si no

$a_{ij} \leftarrow 3i - j^2$

para i  $\leftarrow$  1, n

para j  $\leftarrow$  1, n

escriba a<sub>ij</sub>

Final

b. Datos: 3, 2, 5, 4, 7, 8, 9, 3

Inicio

lea m, n

para i  $\leftarrow$  1, m

para j  $\leftarrow$  1, n

lea a<sub>ij</sub>

may  $\leftarrow$  a<sub>11</sub>

f  $\leftarrow$  1

c  $\leftarrow$  1

para i  $\leftarrow$  1, m

para j  $\leftarrow$  1, n

si |a<sub>ij</sub>|  $\geq$  may

may  $\leftarrow$  |a<sub>ij</sub>|

f  $\leftarrow$  i, c  $\leftarrow$  j

escriba may, f, c, a<sub>ij</sub>

Final

c. Datos: 3, 4, 1, 2, 3, 4, 0, 1, 2, 3, -2, -1, 0, 1

Inicio  
.. cc  $\leftarrow$  0  
.. lea m, n  
.. rm  $\leftarrow \sim (m \geq 1 \vee m \geq 10)$   
.. rn  $\leftarrow \sim (n \geq 1 \vee n \geq 10)$   
.. si rm  $\wedge$  rn = 1 escriba "Sin sentido"  
.. si no  
.... para i  $\leftarrow$  1, m  
..... para j  $\leftarrow$  1, n  
..... lea a<sub>ij</sub>  
..... si a<sub>ij</sub> = 0 cc  $\leftarrow$  cc + 1  
.... escriba cc

Final

d. Datos: 5,4

Inicio  
Lea m  
Lea n  
Para i  $\leftarrow$  1, m  
    p  $\leftarrow$  1;  
    Para j  $\leftarrow$  1, n  
        a[i][j]  $\leftarrow (-1)^{(2i-j)}$   
        si a[i][j] = 0  
            j  $\leftarrow$  n  
            p = 0;  
        si no p  $\leftarrow a[i][j] * p$   
    escriba p

Final

e.

```

Inicio
    n[ 3 ][ 3 ]  $\leftarrow \{ \{60, 70, 40\}, \{40, 50, 70\}, \{70, 30, 90\} \}$ 
    para i  $\leftarrow 1, 3$ 
        f[ i ]  $\leftarrow 0$ 
        para j  $\leftarrow 1, 3$ 
            f[ i ]  $\leftarrow f[ i ] + n[ i ][ j ]$ 
            escriba f[ i ] / 3
    para j  $\leftarrow 1, 3$ 
        c[ j ]  $\leftarrow 0$ 
        para i  $\leftarrow 1, 3$ 
            c[ j ]  $\leftarrow c[ j ] + n[ i ][ j ]$ 
            escriba c[ j ] / 3

```

Final

f. Datos

Inicio

```

    a[3][5]  $\leftarrow \{ \{1, 2, 3, -4, 1\}, \{2, -2, -1, 0, 0\}, \{3, 5, -1, 0, 1\} \}$ 
    salida (a[ ][ ], 3, 5)
    escriba analisis(a[ ][ ], 3, 5)

```

Final

Salida( b [ ][ ], m, n)

Inicio

```

    p  $\leftarrow 1;$ 
    para i  $\leftarrow 1, m$ 
        para j  $\leftarrow 1, n$ 
            escriba b[i][j], " "
            escriba cambio de línea

```

Final

Analisis ( b[ ][ ], m, n)

Inicio

```

    para i  $\leftarrow 1, m$ 
    | para j  $\leftarrow 1, n$ 
        si b[i][j] < 0 & b[i][j] > -3
            r  $\leftarrow i$ 
            c  $\leftarrow j$ 
            i  $\leftarrow m$ 
            j  $\leftarrow n$ 
        si no
            p  $\leftarrow p * b[i][j]$ 

```

escriba r, c

retorne p

Final

### 3.- Arreglos Tridimensionales

- a) Realice la prueba de escritorio y codifíquelo en C++

Datos: 3, 2, 3, 4, 6, 2, 0, 7, 4, 5, 2, 0, 5, 6, 1, 0, 0, 2, 1, 0, 3

Inicio

Haga

Lea A, L, P

Mientras( 0<A<=5 ) y (0<L<=10) y (0<P<=8)

Para i=1, i<=A; i=i+1

Para j=1, j<=L; j=j+1

Para k=1, k<=P; k=k+1

Lea Y[ i ][ j ][ k ]

Para i=1, i<=A; i=i+1

R[ i ] ← fun1(Y, i, L, P)

Para i=1, j<=A; i=i+1

Si R[ i ] = 0

Escriba i

Final

fun1(X, i, L, P )

Inicio

C← 0

Para k=1, k<=P; k=k+1

Para j=1, j<=L; j=j+1

Si (X[ i ][ j ][ k ] ≠ 0 )

C ← C + X[ i ][ j ][ k ]

retornar C

Final

- b) 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.

En el caso de haber aulas disponibles

- 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.

Universidad Nacional Autónoma de Honduras  
Departamento de matemática aplicada  
Programación I (MM-314)  
Guía de ejercicios segundo parcial



septiembre de 2018

### §Funciones

1. Elabore una función que reciba un número entero e imprima la factorización de este número.
2. Escriba una función que reciba un número natural y lo transforme a la base binaria. El prototipo de la función debe ser de la forma

*void binario(int &)*

3. Escriba una función que determine si un número entero es *capicúa*.
4. El  $n$ -ésimo número de Mersenne se define como el número entero  $M_n = 2^n - 1$ . Desarrolle una función que determine si el  $n$ -ésimo número de Mersenne es primo.
5. Escribir una función que calcule el promedio de  $n$  números ingresados por el usuario.
6. Implementar una función que imprima los números enteros entre  $a$  y  $b$  que verifican la propiedad de que la suma de sus dígitos elevados a la  $n$  es igual al mismo número. El prototipo de la función debe ser: *void Funcion(int a, int b, int n)*
7. Implemente la función *máximo común divisor* de forma recursiva, basando se en la siguiente propiedad del máximo común divisor.  
*Si  $a > b$  entonces  $m.c.d(a,b)=m.c.d(a-b,b)$*   
*Si  $a < b$  entonces  $m.c.d(a,b)=m.c.d(a,b-a)$*   
*Si  $a = b$  entonces  $m.c.d(a,b)=m.c.d(b,a)=a=b$*
8. Escriba una programa que aproxime el número  $\pi$  usando la siguiente fórmula:

$$\frac{2}{\pi} = \frac{\sqrt{2}}{2} * \frac{\sqrt{2 + \sqrt{2}}}{2} * \frac{\sqrt{2 + \sqrt{2 + \sqrt{2}}}}{2} * \frac{\sqrt{2 + \sqrt{2 + \sqrt{2 + \sqrt{2}}}}}{2} * \dots$$

9. Una partición de un número natural  $n$  es una forma en la cual  $n$  se puede separar como suma de números enteros positivos. Por ejemplo, para  $n = 4$  se tienen todas sus posibles particiones en listadas:

- $4=1+1+1+1$
- $4=1+2+1$
- $4=2+2$
- $4=3+1$
- $4=4$

A partir de lo anterior se define el número de particiones totales de  $n$  y se denota como  $p(n)$ . En el ejemplo anterior obtendríamos que  $p(4) = 5$ . Adicionalmente a lo anterior se define  $\pi_m(n)$  como el número de particiones de  $n$  en el cual ningún sumando es más grande que  $m$ . Por ejemplo  $\pi_2(5) = 3$  dado que:

- $5=1+1+1+1+1$
- $5=2+2+1$
- $5=2+1+1+1$

Existe un resultado en la teoría de números que establece que

$$\pi_m(n) = \pi_m(n - m) + \pi_{m-1}(n)$$

donde  $\pi_m(n) = 1$  si es  $n = 0$  y 0 si  $n < 0$ . Además definimos  $\pi_0(n) = 0$ . Con ayuda de lo anterior escriba una función recursiva que calcule  $p(n)$ .

10. Escriba una función que reciba tres arreglos de tamaño 3, representantes de 3 vectores del espacio tridimensional y determine el volumen del paralelepípedo formado por estos tres vectores.
11. Crear una función que reciba un número real, el programa deberá imprimir tal número con su desglose ideal en la moneda local. Por ejemplo, si la función recibe 1345.56, entonces se deberá imprimir: 2 billetes de 500, 3 billetes de 100, 2 billetes de 20, 2 billetes de 2, 1 billete de 1, 1 moneda de 5 centavos, 1 moneda de 1 centavo.
12. Suponga que 5 amigos participan en una quiniela futbolística. Las apuestas se hacen sobre el grupo F del mundial en Rusia 2018. Cada uno de los interesados hace sus apuestas previamente sobre cada uno de los seis partidos. A continuación se siguen las siguientes reglas: 3 puntos si acierta a un empate o a un gane, 3 puntos extra si se le atina al resultado con goles y 0 puntos de lo contrario. Cree un programa que pida las apuestas inicialmente, luego el programa deberá generar aleatoriamente los resultados reales de los 6 encuentros (suponga que ningún equipo anota más de 6 goles) para luego mostrar una tabla de los puntos de los miembros de la quiniela.
13. Repita el ejercicio anterior, pero ahora simule las apuestas y use los datos reales de los resultados del grupo F.
14. Se dice que  $n$  y  $m$  son primos relativos si no tienen divisores mayores que 1 en común. Por ejemplo 4 y 21 son primos relativos, puesto que no tienen divisores en común, por otro lado, 4 y 34 no son primos relativos puesto que tienen al 2 como divisor común. Con lo anterior se define  $\phi(n)$  como el número de enteros que son primos relativos con  $n$  no mayores que  $n$ . Por ejemplo,  $\phi(8) = 4$  puesto que los números con los que 8 conforma primos relativos son 1, 3, 5 y 7. Existe un resultado en matemáticas que establece que  $\phi(n)/n$  se puede escribir como el producto de todos los factores que tienen la forma  $\left(1 - \frac{1}{p}\right)$  donde  $p$  es un número primo divisor de  $n$ . Use el resultado anterior para desarrollar una función que calcule  $\phi(n)$ .

15. Al ejecutar este programa obtenemos como salida de pantalla:

```
1 #include<iostream>
2 #include <cstdlib>
3 using namespace std ;
4 void Rec();
5 int main (){
6     Rec();
7     return 0;
```

```
8 void Rec(){
9     int x=rand()%2;
10    if(x==0)
11        cout<<"Sali del laberinto"<<
12    else
13        Rec();}
```

A. Sali del laberinto

B. No imprime nada.

16. Al ejecutar este programa obtenemos como salida de pantalla:

```
1 #include<iostream>
2 #include <cstdlib>
3 using namespace std ;
4 void MD(int [], int [], int );
5 int main (){
6     int X[5]={1,4,2,1,2};
7     int C[5];
8     MD(X,C,5);
9     return 0;
10 void MD(int X[], int C[], int N){
```

```
11     for(int i=0;i<N; i++){
12         C[i]=1;
13     for(int i=0;i<N; i++){
14         for(int j=0;j<N; j++){
15             if(i!=j){
16                 if(X[i]==X[j])
17                     C[j]++;
18             }
19         cout<< " " <<C[i];}}
```

A. 2 1 2 2 2

B. 1 2 2 2 1

C. 2 1 2 1 2

D. 2 2 1 2 2

17. Despues de ejecutarse la linea 9, ¿Cuál es el valor de B[3][2]?

```
1 #include<iostream>
2 using namespace std ;
3 void Potencia(int [][]4, int [][]4) ;
4 int main( ){
5     int B [4][4], A [4][4];
6     for(int i=0; i<4; i++)
7         for(int j=0; j<4; j++)
```

```
8         A[i][j]=i;
9     Potencia(A,B);
10    return 0;
11 void Potencia(int A[][4], int B[][4]){
12     for(int i=0; i<4; i++)
13         for(int j=0;j<4;j++)
14             B[i][j]=A[i][i]*A[j][j];}
```

A. 2

B. 12.

C. 18.

D. 6.

18. Al ejecutar este programa obtenemos como salida de pantalla:

```
1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4 void Fd(int [], int , int , int );
5 int main(){
6     int B[5];
7     for(int i=0;i<5;i++)
8         B[i]=pow(i,4-i);
9     Fd(B,3,0,5);
```

```
10    return 0;
11 void Fd(int B[], int x, int k, int N)
12 {
13     if(k<N){
14         if(B[k]==x)
15             cout<<"Fd con "<<k<<endl;
16         else
17             return Fd(B,x,k+1,N);}}
```

A. Fd con 1

B. Fd con 4

C. Fd con 3

D. Fd con 0

19. ¿Qué se imprime en la linea 6?

```
1 #include<iostream>
2 using namespace std;
3 double mid(double [], int , int );
4 int main(){
```

```

5     double A[6] = { -2.1, 3.2, 5.4, 5.5, 12.3, 7.1, 13.2 };
6     cout << "La mid es: " << mid(A, 0, 5) << endl;
7     return 0;
8 double mid(double A[], int n, int m) {
9     if ((m-n+1)%2!=0)
10        return A[(m-n+1)/2];
11    else
12        return (A[(m-n+1)/2]+A[(m-n+1)/2-1])/2;
}

```

A. La mid es: 54.5

B. La mid es: 54

C. La mid es: 55

D. La mid es: 28.6

## §Arreglos

- 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 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 2 2 2 2 2 2 1	1 2 2 2 2 2 2 2 2 1
1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1	1 2 3 3 3 3 3 2 1
1 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 2 1	1 2 3 4 4 4 4 3 2 1
1 2 3 4 5 5 5 5 5 5 5 5 5 5 5 5 4 3 2 1	1 2 3 4 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 5 4 3 2 1
1 2 3 4 5 6 7 7 7 7 7 7 7 6 5 4 3 2 1	1 2 3 4 5 6 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 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 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 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 7 6 5 4 3 2 1
1 2 3 4 5 6 7 7 7 7 7 7 7 6 5 4 3 2 1	1 2 3 4 5 6 7 7 7 7 6 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 6 6 6 6 5 4 3 2 1
1 2 3 4 5 5 5 5 5 5 5 5 5 5 5 4 3 2 1	1 2 3 4 5 5 5 5 4 3 2 1
1 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 2 1	1 2 3 4 4 4 4 4 3 2 1
1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1	1 2 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 2 2 2 1	1 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 1 1 1 1	1 1 1 1 1 1 1 1 1

$N = 10$

$N = 5$

- Use un arreglo bidimensional de tamaño  $10 \times 10$  para almacenar los primeros 100 números primos y visualice esta tabla por pantalla con formato de tabla.
- Escriba un programa que genere 1000 enteros de forma aleatoria entre 0 y 10, luego debe generarse una tabla que contenga la frecuencia con la que aparece cada entero.
- Suponga que se tienen  $n$  jarras rojas y  $n$  jarras azules de jugo, todas de diferentes formas y tamaños. Todas las jarras rojas pueden contener diferente cantidad de jugo, de igual forma las azules. Más aun, para cada jarra roja, existe una jarra azul que tiene la misma capacidad y viceversa. Imagina que quieres agrupar las jarras en parejas(roja con azul) de forma que tengan la misma capacidad. Para hacer esto se admite la siguiente operación: agarra una pareja de jarras, una azul y otra roja, llena completamente la jarra roja con agua y entonces derrama el agua de esta jarra en la jarra azul. Hacer esto te debería indicar si las jarras contienen la misma cantidad de agua o no. Escriba un programa

que por medio de mensajes de pantalla le indique a alguien que hacer para terminar *esta* tarea.

5. Crear un programa que contenga un arreglo de números enteros que se solicitan al usuario, a continuación se imprimirán aquellos elementos de este arreglo que sean múltiplos de 5 acompañados de su índice respectivo en el arreglo.
6. Crear un arreglo de enteros unidimensional de tamaño 25 que contenga los primeros 25 números primos de Mersenne(Ejercicios 4 de funciones).
7. Desarrolle un programa que almacene 10 números dados por el usuario en un arreglo de tamaño 10. Al final el programa deberá imprimir el mayor de los valores ingresados, el menor y el promedio.

### §Ejercicios Misceláneos

1. Escriba una función que reciba un arreglo de números reales y determine cual es el elemento que más se repite. En estadística este dato se conoce con el nombre de *moda*.
2. Escriba una función que reciba dos arreglos de enteros de tamaño 20 y un número entero positivo  $n$ . La función debe almacenar en un arreglo los dígitos de  $n$  que sean pares y en el otro los impares.
3. Escriba un programa que trabaje con un arreglo de caracteres bidimensional de tamaño  $8 \times 8$  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.
4. Escriba una función que reciba dos arreglos de enteros y determinen si uno de ellos contiene todos los elementos que contiene el otro.(Asuma que en los arreglos los elementos no se repiten.)
5. Escriba una función que reciba un arreglo unidimensional de forma que se interprete como un vector. A continuación la función debe devolver la norma de este vector.
6. Escriba una función que reciba un arreglo bidimensional de tamaño  $4 \times 4$  de forma que se interprete como una matriz, luego la función debe retornar el determinante de esta matriz.
7. Escriba una función que reciba dos arreglos unidimensionales de modo que se interpreten como dos vectores, a continuación la función deberá imprimir la distancia entre los dos vectores.
8. Escriba una función que calcule la matriz inversa de una matriz de tamaño  $4 \times 4$  usando el determinante. Para lo anterior la función recibirá 2 arreglos bidimensionales de tamaño  $4 \times 4$ .
9. Implemente una función que determine los índices del elemento con valor mínimo en una tabla de números reales.
10. Implemente una función que determine el valor que más se repite en una tabla de números reales.

11. Suponga que una mano de cartas es representada por un arreglo bidimensional de tamaño  $2 \times 5$ , 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.
12. Desarrolle una función que reciba un arreglo unidimensional de tamaño 3, este arreglo representara un triple tiro de un dado. La función deberá asignar a este arreglo valores que represente el triple tiro.
13. Suponga que dos arreglos enteros tienen el mismo tamaño. Implemente una función que determine si los arreglos contienen los mismo elementos.
14. Suponga que un número decimal positivo se representa por medio de un arreglo. El punto decimal se representa con el número -1, luego los números a lado izquierdo representan la parte entera y los números en la parte derecha representan la parte decimal. Por ejemplo los siguientes arreglos
- |    |   |   |   |   |   |
|----|---|---|---|---|---|
| -1 | 1 | 5 | 3 | 9 | 1 |
|----|---|---|---|---|---|
- |   |   |    |   |   |   |
|---|---|----|---|---|---|
| 3 | 4 | -1 | 2 | 0 | 5 |
|---|---|----|---|---|---|
- |   |   |   |   |   |    |
|---|---|---|---|---|----|
| 1 | 2 | 4 | 5 | 6 | -1 |
|---|---|---|---|---|----|
- representan respectivamente los números decimales 0.15391, 34.205 y 12456.
- Desarrolle una función que reciba un número real en este formato e imprima la raíz cuadrada de este número usando el *método largo del cálculo de raíces cuadradas*.
15. 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 utilizó 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  $30 \times 10$  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 número 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  $(2, 3, 4, \dots, 12)$  y su respectiva frecuencia.
- Ordenar e imprimir el arreglo  $FR$  de menor a mayor.

16. Suponga que se tiene un arreglo bidimensional de  $4 \times 4$ , 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 los 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.

17. Implemente la función `DividirParidad`. Esta tiene como prototipo de función:

```
void DividirParidad(int A[], int N)
```

La función recibe un arreglo de enteros de tamaño  $N$  para luego asignar a cada elemento de este, números aleatorios entre 1000 y 10000. Posteriormente la función debe colocar al inicio del arreglo todos los números pares que este contenga.

18. Implemente una función que reciba un arreglo  $A$  y un número real  $c$ . Los elementos del arreglo en este ejercicio representan los coeficientes de un polinomio, es decir, si  $A = < a_0, a_1, \dots, a_n >$  entonces este arreglo representa al polinomio  $p(x) = a_0 + a_1x + \dots + a_nx^n$ . El retorno de esta función debe ser el polinomio evaluado en  $c$ , es decir  $p(c)$ .
19. Implemente una función que reciba tres arreglos  $A$ ,  $B$  y  $C$  de tamaños  $n$ ,  $m$  y  $m+n$  respectivamente(use constantes para este ejercicio). Como en el ejercicio anterior los arreglos representan polinomios, en este sentido la función debe asignar al arreglo  $C$  los coeficientes de la multiplicación de los polinomios representados por  $A$  y  $B$ .
20. Implemente una función que reciba cuatro arreglos  $A$ ,  $B$ ,  $C$  y  $R$  de tamaños  $n$ ,  $m$ ,  $n-m$  y  $m-1$  respectivamente(use constantes para este ejercicio). Como en el ejercicio anterior los arreglos representan polinomios, en este sentido la función debe asignar al arreglo  $C$  y  $R$  los coeficientes del cociente entre  $A$  y  $B$  además de el residuo de  $A$  entre  $B$ .
21. Implemente una función que reciba como argumentos dos arreglos  $A = < a_1, \dots, a_n >$  y  $B = < b_1, \dots, b_m >$ . La función debe imprimir el máximo de las diferencias de los valores absolutos de los elementos de ambos arreglos.
22. Implemente una función que reciba un arreglo de números reales y determine la mediana estadística de los elementos de este arreglo.
23. Implemente una función que tenga el siguiente prototipo:

```
void TablaFrecuencia(double A[], int N, int C)
```

La función deberá imprimir una clásica tabla de datos estadísticos de los elementos de  $A$  con dividido en  $C$  clases. La tabla deberá contener los siguientes datos: *Clases, frecuencias, marcas de clase, frecuencia acumulada*.

24. Generar aleatoriamente un arreglo de enteros positivos de tamaño 300, luego cree una función que reciba este arreglo y devuelva la cantidad de elementos de este arreglo que son primos. Repita el experimento anterior 30 veces, ¿A qué conclusión puede llegar?
25. Escriba una función que reciba un arreglo de caracteres y un número  $m$ , la función deberá mover cada elemento del arreglo  $m$  posiciones hacia adelante, de forma que al rebasar el tamaño del arreglo regrese a las primeras posiciones.
- 26 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 1 y 2 mientras que Licien tira 3 y 4, 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.

27. 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 {1, 2, 3}, entonces si observamos todas las posibles combinaciones {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3} notamos que las únicas combinaciones que suman un múltiplo de tres son {1, 2} y {2, 3}, de esta forma en este tiro el jugador acumularía  $1+2=3$  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.

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.

- (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)
```

28. 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 esté 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

29. Suponga que en un cierto país se tienen monedas con denominaciones 2, 3, 5, 7, 11 y 13, Tres amigos suyos y usted tienen cada uno una bolsa con muchas monedas de todas las denominaciones existentes y están dispuestos a jugar un juego llamado *monedas divisoras*, el juego se divide en 5 partidas, en cada partida cada uno de los jugadores escoge en secreto una moneda de su bolsa y simultáneamente desvelan la moneda escogida tirándola al piso en el centro de los jugadores, a continuación se procede a efectuar la suma de los valores de las monedas en juego, los jugadores obtienen puntos compartidos si el valor de la moneda jugada divide a la suma de todas ellas, si existe solo un jugador con esta última propiedad, el jugador se lleva 4 puntos, si hay dos que verifican esta propiedad cada uno se lleva dos puntos, si hay tres o cuatro jugadores que cumplen la condición entonces cada uno se anotará 1 punto; al finalizar la quinta partida se suman los puntos obtenidos por cada jugador y se escoge al ganador como el jugador que más puntos hizo. Simule el juego en C++ usando la función *rand*. Para ello haga lo que se le pide a continuación.

- (a) Implemente una función con el siguiente prototipo

```
void Generar(int M[ ], int J[ ], int P[ ])
```

En esta función,  $M$  es un arreglo de tamaño 6 que contiene las denominaciones de las monedas (este arreglo no se modifica),  $J$  es el arreglo que contendrá las denomi-

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.

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.

naciones generadas por la función `rand` de las monedas tiradas por los 4 jugadores y en  $P$  se irá acumulando la puntuación de cada jugador de acuerdo a las reglas.

- (b) Simule las 5 partidas en el programa principal con ayuda de la función `Generar`, imprimiendo finalmente un ganador.

## §Ejercicios de investigación

1. **Coordenadas polares.** Escriba una función que transforme de coordenadas rectangulares a coordenadas polares.
2. **Base binaria.** Escribir una función que transforme un número en base 10 a base binaria.
3. **Tic tac toe.** Simule el juego de tie tac toe con un arreglo bidimensional de tamaño  $3 \times 3$ .
4. **Batalla Naval.** Escriba un programa que simule el juego de la batalla naval para jugar contra la computadora.
5. **Ahorcado.** Escriba un programa que simule el juego del ahorcado.
6. **Búsqueda binaria** Desarrolle una función que implemente la búsqueda binaria en un arreglo unidimensional.
7. **Ordenamiento Quicksort.** Implemente el algoritmo de ordenamiento `Quicksort`.
8. **Ordenamiento Selección** Implemente el algoritmo de ordenamiento de `Selección`.
9. **Producto interior.** Implemente una función que reciba dos arreglos unidimensionales de forma que estos representen dos vectores en  $\mathbb{R}^n$ . La función debe retornar el producto interior de los dos vectores.
10. **Eliminación Gaussiana.** Implemente una función que reciba una matriz(arreglo bidimensional)  $A$  de tamaño  $10 \times 10$  y un dos vectores(arreglos unidimensionales )  $x$  y  $b$  de tamaño 10. La función debe almacenar en  $x$  la solución del sistema  $Ax = b$  usando el método de la eliminación Gaussiana.
11. **Torres de Hanoi.** Implemente una función que imprima los pasos para resolver el problema de las *torres de Hanoi*.
12. **Calendario** Investigue inicialmente un algoritmo que calcule el día de la semana de una fecha dada. A continuación escriba una función que calcule el día de la semana dada la fecha usando el algoritmo que investigo.  
Investigue por otro lado cuando un año es bisiesto.  
Lo anterior es suficiente para hacer la siguiente tarea: Desarrolle una función que reciba un año y un mes, la función deberá imprimir el calendario de ese mes en ese año de forma convencional.
13. **Mínimo común múltiplo** Investigue como se puede escribir el mínimo común múltiplo de dos números en función del máximo común divisor de estos. Con lo anterior desarrolle una función que calcule el mínimo común múltiplo de dos números enteros.