

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 π 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 . Po 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 defina se $\pi_0(n) = 0$

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 50 centavos, 1 moneda de 5 centavos y 1 moneda de 1 centavos.
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"<<endl;
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     for (int i=0;i<N;i++){
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. Después de ejecutarse la línea 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 línea 6?

```

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

```

```

5    double A[6]={ -2.1,3.2,54,55,123.71,132};
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

C. La mid es: 55

B. La mid es: 54

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

```

$N = 10$

$N = 5$

- 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.
- 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 jugos, 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 8×8 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 4×4 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 4×4 usando el determinante. Para lo anterior la función recibirá 2 arreglos bidimensionales de tamaño 4×4 .
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 2×5 , 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 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 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 4×4 , 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.

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 $A = \langle a_0, a_1, \dots, a_n \rangle$ 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 $A = \langle a_1, \dots, a_n \rangle$ y $B = \langle b_1, \dots, b_m \rangle$. 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				





David	16	23	43	45
Licien	14	12	52	61

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.

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.

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 , ,  , entonces si observamos todas las posibles combinaciones $\{\langle \rangle\}$, $\{\langle \rangle\}$, $\{\langle \rangle\}$, $\{\langle \rangle\}$, $\{\langle \rangle, \langle \rangle\}$, $\{\langle \rangle, \langle \rangle\}$, $\{\langle \rangle, \langle \rangle\}$ notamos que las únicas combinaciones que suman un múltiplo de tres son $\{\langle \rangle, \langle \rangle\}$ y $\{\langle \rangle, \langle \rangle\}$, 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.

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 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.

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-

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 tic tac toe con un arreglo bidimensional de tamaño 3×3
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 10×10 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.