



Solución Segunda Práctica Calificada

CC102-CD

Ciclo: 2017-1

1. [5 pts.] El mínimo común múltiplo (**mcm**) de 2 números m y n es el resultado de multiplicar todos los factores comunes y no comunes elevados a la mayor potencia, por ejemplo el **mcm** de $10 = 2 \cdot 5$ y $12 = 2^2 \cdot 3$ será $2^2 \cdot 3 \cdot 5 = 60$. Una forma práctica de calcular el mcm es:

int $i = 2$; // posible divisor de m y n
int $mcm = 1$;

m	n	i	mcm
10	12	2	mcm $\rightarrow 2$
5	6	2	mcm $\rightarrow 4$
5	3	2 \rightarrow 3	mcm $\rightarrow 12$
5	1	3 \rightarrow 4 \rightarrow 5	mcm $\rightarrow 60$
1	1	5	Fin: $m==n==1$

Escriba un programa que implemente la función **mcm(...)**.

// cálculo del mínimo común múltiplo

```
#include<stdio.h>
```

```
int mcm(int m, int n);
```

```
void main(void){
```

```
    int m = 10, n = 12;
```

```
    printf("mcm(%d, %d) = %d\n", m, n, mcm(m, n));
```

```
}
```

```
int mcm(int m, int n){
```

```
    int mm = 1, i = 2;
```

```
    while(m!=1 || n!=1){
```

```
        while (m%i==0 && n%i==0) {m /=i; n /=i; mm *=i;}
```

```
        while (m%i==0 )           {m /=i; mm *=i;}
```

```
        while (n%i==0 )           {n /=i; mm *=i;}
```

```
        i++;
```

```
    }
```

```
    return mm;
```

```
}
```

2. [5 pts] Usando una función recursiva, imprima la serie de Fibonacci hasta un número n ingresado.

```
#include <stdio.h>
```

```
long fibonacci(long n);
```

```
void main(void){
```

```
    int i=0,n;
```

```
    printf("Ingrese un numero n mayor o igual a 0: ");
```

```
    scanf("%d",&n);
```

```
    while(i<=n) printf("%lu, ",fibonacci(i++));
```

```
    printf("\n");
```

```
}
```

```
long fibonacci(long n) {
```

```
    if(n<2) return n;
```

```
    return fibonacci(n-1)+fibonacci(n-2);
```

```
}
```

3. [5 pts] Escriba un programa que lea $n > 1$, defina `arr[n]`, lea valores, en el rango [10, 20], para `arr` desde el teclado, los valores pueden ser repetidos en modo consecutivo. Mostrar al final del ingreso, la cantidad de veces que se repite cada valor dentro del arreglo. Ejemplo:

Ingresar n: 6

Ingresar elementos: 10, 10, 10, 19, 18, 18.

Estadística:

Elemento 10 se repite 3 veces.

Elemento 19 se repite 1 vez.

Elemento 18 se repite 2 veces.

Sugerencia: La estadística puede hacerse en tres partes: INICIO para el primer elemento del arreglo; REPETITIVA para todo el arreglo; y FIN para reportar la última estadística.

```
#include<stdio.h>
```

```
void main(void){
```

```
    int n, i, valor, cant;
```

```
    printf("Ingrese un entero > 1: ");
```

```
    scanf("%d", &n);
```

```
    int arr[n];
```

```
    for(i=0; i<n; i++){
```

```
        printf("Ingrese el dato [%d] entre 10 y 20: ", i+1);
```

```
        scanf("%d", &arr[i]);
```

```
    }
```

```
// Estadística
```

```
    valor = arr[0];
```

```
    cant = 1;
```

```
    i=1;
```

```
    while(i<n){
```

```
        if(arr[i]!= valor) {
```

```
            printf("Elemento %d se repite %d veces.\n", valor, cant);
```

```
            cant = 1;
```

```
            valor = arr[i];
```

```
        } else cant++;
```

```
        i++;
```

```
    }
```

```
    printf("Elemento %d se repite %d veces.\n", valor, cant);
```

```
}
```

4. [5 pts.] El sorteo semanal de la *Tinka* premia al apostador que acierta 6 números de un total de 46 bolas, seleccionadas al *azar* desde una urna. Tenga en cuenta que un número que sale sorteado, no se puede volver a repetir. Elabore un programa que contenga los 46 números a sortear en un *arreglo* para luego obtener otro arreglo de los números correspondientes a los sorteados.

```
/* Sorteo de la TINKA*/
```

```
#include<stdio.h>
```

```
#include<time.h>
```

```
#include<stdlib.h>
```

```
#define numBol 46
```

```
#define nSaques 6
```

```
void main(void){
```

```
    int urna[numBol], k=numBol, saque, i, j;    // i = índices
```

```
    for(i=0; i<numBol; i++) urna[i]=i+1;        // llena la urna con bolas
```

```

srand(time(NULL)); j=0;
for(j=1; j<=nSaques; j++){ // Extracción de bolas
    i = rand()%k--; // número random entre 0 y 45, 44, 43, ... 40
    saque = urna[i]; // bola sacada al azar
    urna[i] = urna[k]; // cambia la bola de la cola a la posición i
    urna[k] = saque; // cambia la bola sacada a la cola
}
printf("Números extraídos:\n"); // imprime la cola en orden inverso
for(i=numBol-1; i>=numBol-nSaques;i--) printf("%d\n",urna[i]);
}

```