

---

## Laboratorio 02

### Bucles

**Nombre:** Gabriel Fernando Rodriguez Cutimbo

**CUI:** 20212157

**Grupo:** B

### 1. Ejercicios

Resolver los siguientes ejercicios planteados:

1. Sumar todos los enteros pares desde 2 hasta 100.

**Código:**

```
#include <iostream>
using namespace std;

int main(){
    //Memoria
    int sum=0;

    //Proceso de suma
    for(int i=2; i<=100; i+=2){
        sum+=i;
    }
    cout<<"La suma de los enteros pares desde 2 hasta 100 es: "<<sum;

    return 0;
}
```

**Funcionamiento:**

```
La suma de los enteros pares desde 2 hasta 100 es: 2550
```

2. Calcule los primeros 50 números primos y muestre el resultado en pantalla.

**Código:**

```
#include <iostream>
using namespace std;

int main(){
    //Memoria
    const int n=50;
    int it=0;
    int prim[n];
    int tempn=0;
```

```
bool prueba = true;

//Proceso de búsqueda de primos
while(it<n){
    // Casos especiales
    if (tempn==0||tempn==1||tempn==4){
        prueba = false;
    }
    for (int x=2; x<tempn/2; x++){
        if (tempn%x==0){
            prueba = false;
        }
    }
    if(prueba==false){
        tempn++;
    }
    // Si no se pudo dividir por ninguno de los de arriba, sí es primo
    if(prueba==true){
        prim[it]=tempn;
        tempn++;
        it++;
    }
    prueba=true;
}

//Mostrar vector
for(int i=0; i<n; i++){
    cout<<prim[i]<<" ";
}

return 0;
}
```

**Funcionamiento:**

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109
113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223 227 229
```

3. Escribir un programa que visualice en pantalla los números múltiplos de 5 comprendidos entre 1 y 100.

**Código:**

```
#include <iostream>
using namespace std;

int main(){
    //Proceso
    for(int i=0; i<=100; i+=5){
        cout<<i<<" ";
    }

    return 0;
}
```

**Funcionamiento:**

0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

4. Escriba un código que solicite ingresar dos números  $x$  y  $y$ , tal que  $x < y$ . Muestre todos los números primos que se encuentren entre el rango de valores, de no encontrarse, mostrar el primo más cercano a  $x$  o  $y$ .

**Código:**

```
#include <iostream>
using namespace std;

int main(){
    //Memoria 1
    int it=0;
    int tempn;
    bool prueba = true;

    //Memoria 2
    int a, b;
    do{
        cout<<"Ingrese el primer número (este debe ser menor que el siguiente número): ";
        cin>>a;
        if(a<0){
            cout<<"Error, intentelo de nuevo";
        }
    } while(a<0);
    do{
        cout<<"Ingrese el segundo número: ";
        cin>>b;
        if(a>=b||b<0){
            cout<<"Error, intentelo de nuevo";
        }
    } while(a>=b||b<0);
    int n=b-a;
    int* prim = new int[n];
    //Todos los elementos del vector seran 0
    for(int i=0; i<n; i++){
        prim[i]=0;
    }

    //Proceso de búsqueda de primos
    tempn=a;
    for(int i=0; i<n; i++){
        // Casos especiales
        if (tempn==0||tempn==1||tempn==4){
            prueba=false;
        }
        for (int x=2; x<tempn/2; x++){
            if (tempn%x==0){
                prueba=false;
            }
        }
        if(prueba==false){

```

```
        tempn++;
    }
    // Si no se pudo dividir por ninguno de los de arriba, sí es primo
    if(prueba==true&&tempn>=a&&tempn<=b){
        prim[it]=tempn;
        tempn++;
        it++;
    }
    prueba=true;
}

//Mostrar vector
for(int i=0; i<n; i++){
    if(prim[i]!=0){
        cout<<prim[i]<<" ";
    }
}

delete[] prim;

return 0;
}
```

**Funcionamiento:**

```
Ingrese el primer número (este debe ser menor que el siguiente número): 10
Ingrese el segundo número: 20
11 13 17 19
```

5. Elabore un programa que lea  $n$  números y determine cuál es el mayor, el menor y la media de los números leídos.

**Código:**

```
#include <iostream>
using namespace std;

int main(){
    //Memoria 1
    int n;
    cout<<"Ingrese la cantidad de números que aceptará este programa: ";
    cin>>n;
    int* vec = new int[n];
    for(int i=0; i<n; i++){
        cout<<"Ingrese un número: ";
        cin>>vec[i];
    }

    //Memoria 2
    int max=vec[0];
    int min=vec[0];
    int med=0;

    //Número mayor
```

```
for(int i=0; i<n; i++){
    if(max<vec[i]){
        max=vec[i];
    }
}

//Número menor
for(int i=0; i<n; i++){
    if(min>vec[i]){
        min=vec[i];
    }
}

//La media de todos los números
for(int i=0; i<n; i++){
    med+=vec[i];
}
med=med/n;

//Salida
cout<<"El mayor número de todos los números es: "<<max<<endl;
cout<<"El menor número de todos los números es: "<<min<<endl;
cout<<"La media de todos los números es: "<<med<<endl;

delete[] vec;
return 0;
}
```

**Funcionamiento:**

```
Ingrese la cantidad de números que aceptará este programa: 5
Ingrese un número: 12
Ingrese un número: 13
Ingrese un número: 11
Ingrese un número: 9
Ingrese un número: 1
El mayor número de todos los números es: 13
El menor número de todos los números es: 1
La media de todos los números es: 9
```

6. Elabore un programa que calcule la serie de Fibonacci. La serie de Fibonacci es la sucesión de números: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... Cada número se calcula sumando los dos anteriores a él.

**Código:**

```
#include <iostream>
using namespace std;

int main(){
    //Memoria
    int nveces,ant,sig,sum;
    ant=0;
```

```
sig=1;

//Fibonacci
cout<<"Este programa genera la sucesión de Fibonacci." <<endl;
cout<<"Cuantas nveces se generara la sucesión de Fibonacci: ";
cin>>nveces;
if(nveces==1){
    cout<<ant;
}
if(nveces>=2){
    cout<<ant<<" "<<sig;
}
for(int n=3;n<=nveces;n++){
    sum=ant+sig;
    cout<<" "<<sum;
    ant=sig;
    sig=sum;
}

cout<<endl;

return 0;
}
```

**Funcionamiento:**

```
Este programa genera la sucesión de Fibonacci.
Cuantas nveces se generara la sucesión de Fibonacci: 10
0, 1, 1, 2, 3, 5, 8, 13, 21, 34
```

7. Calcula el promedio de 3 notas para n estudiantes.

**Código:**

```
#include <iostream>
using namespace std;

int main(){
    //Memoria
    int a[3];
    int prom=0;

    //Promedio
    cout<<"Este programa calcula el promedio de 3 notas"<<endl;
    for(int i=0; i<3; i++){
        cout<<"Ingrese la nota: ";
        cin>>a[i];
        prom+=a[i];
    }
    prom=prom/3;

    //Salida
    cout<<"El promedio de las 3 notas es: "<<prom;

    return 0;
}
```

}

**Funcionamiento:**

```
Este programa calcula el promedio de 3 notas
Ingrese la nota: 10
Ingrese la nota: 20
Ingrese la nota: 30
El promedio de las 3 notas es: 20
```

8. Escribir un programa que genere la tabla de multiplicar de un número introducido por el teclado.

**Código:**

```
#include <iostream>
using namespace std;

int main(){
    //Memoria
    int n=0;

    //Ingreso de datos
    while(n<=0){
        cout<<"Ingrese el número hasta donde se mostrara la tabla de multiplicar: ";
        cin>>n;
        if(n<=0){
            cout<<"Error vuelva a intentarlo"<<endl;
        }
    }
    cout<<endl;

    //Reservación de memoria para la matriz m1
    int** m1 = new int* [n];
    for(int i=0; i<n; i++){
        m1[i] = new int [n];
    }

    //Lector de datos de la matriz m1
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            m1[i][j]=(i+1)*(j+1);
        }
    }
    cout<<endl;

    //Mostrar la interfaz horizontal superior
    cout<<"Tabla de multiplicación del 1 hasta el "<<n<<endl;
    for(int i=0; i<=n; i++){
        if (i==0){
            cout<<"*\t\t";
        }
        else{

```

```

        cout<<i<<"\t";
    }
}
cout<<endl;
for(int i=0; i<=n+1; i++){
    cout<<"-----";
}
cout<<endl;

//Mostrador de datos de la matriz m1 // Mostrar la interfaz vertical izquierda
for(int i=0; i<n; i++){
    cout<<m1[i][0]<<"\t\t";
    for(int j=0; j<n; j++){
        cout<<m1[i][j] << "\t";
    }
    cout<<endl;
}
cout<<endl;

//Liberación de memoria para la matriz m1
for(int i=0; i<n; i++){
    delete[] m1[i];
}
delete[] m1;

return 0;
}

```

### Funcionamiento:

```

Ingrese el número hasta donde se mostrara la tabla de multiplicar: 5

Tabla de multiplicación del 1 hasta el 5
*   |   1   2   3   4   5
-----
1   |   1   2   3   4   5
2   |   2   4   6   8  10
3   |   3   6   9  12  15
4   |   4   8  12  16  20
5   |   5  10  15  20  25

```

9. Escribir un programa que pida al usuario un número entero y muestre por pantalla un triángulo rectángulo como el de más abajo, de altura el número introducido.

```

*
**
***
****
*****

```

### Código:

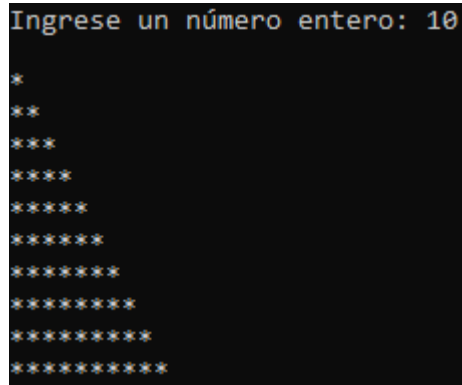


```
#include <iostream>
using namespace std;

int main(){
    //Memoria
    int temp=1;
    int n;
    cout<<"Ingrese un número entero: ";
    cin>>n;
    cout<<endl;
    if(n<0){
        n=n*-1;
    }

    //Triángulo Rectángulo
    for(int i=0; i<n; i++){
        for(int j=0; j<temp; j++){
            cout<<"*";
        }
        temp++;
        cout<<endl;
    }

    return 0;
}
```

**Funcionamiento:**

10. Escribir un programa que pida al usuario una palabra y luego muestre por pantalla una a una las letras de la palabra introducida empezando por la última.

**Código:**

```
#include <iostream>
#include <string>
using namespace std;

int main(){
    //Memoria
    string palabra;
    cout<<"Ingrese una palabra: ";
    getline(cin,palabra);

    //Reverso
```

```
cout<<"Esta seria la palabra invertida: ";  
for(int i=(palabra.size()-1); i>=0; i--){  
    cout<<palabra[i];  
}  
  
return 0;  
}
```

### **Funcionamiento:**

```
Ingrese una palabra: UNSA  
Esta seria la palabra invertida: ASNU
```

## 2. Entregables

Al final estudiante deberá:

- 2.1.** Compactar el código elaborado y subirlo al aula virtual de trabajo. Agregue sus datos personales como comentario en cada archivo de código elaborado.
- 2.2.** Elaborar un documento que incluya tanto el código como capturas de pantalla de la ejecución del programa. Este documento debe de estar en formato PDF.
- 2.3.** El nombre del archivo (comprimido como el documento PDF), será su LAB02\_GRUPO\_A/B/C\_CUI\_1erNOMBRE\_1erAPELLIDO.  
(Ejemplo: LAB02\_GRUPO\_A\_2022123\_PEDRO\_VASQUEZ).
- 2.4.** Debe remitir el documento ejecutable con el siguiente formato:  
LAB02\_GRUPO\_A/B/C\_CUI\_ EJECUTABLE\_1erNOMBRE\_1erAPELLIDO  
(Ejemplo: LAB02\_GRUPO\_A\_EJECUTABLE\_2022123\_PEDRO\_VASQUEZ).

En caso de encontrarse trabajos similares, los alumnos involucrados no tendrán evaluación y serán sujetos a sanción.