Laboratorio Clase 10

- Practico de Laboratorio

REVISION EJERCICIOS

Capitulo 11 – Sección 11.5 - Ejercicio 6

Escriba un programa que declare tres arreglos unidimensionales llamados voltios, corriente, resistencia. Cada arreglo deberá declararse en main() y deberá ser capaz de contener 10 números de precisión doble. Los números que deberán almacenarse en corriente son:

10.62 – 14.89 – 13.21 – 16.55 – 18.62 - 9.47 - 6.58 – 18.32 – 12.15 – 3.98

Los números que deberán almacenarse en resistencia mediante ingreso interactivo (cin) serán:

4 – 8.5 – 6 – 7.35 – 9 – 15.3 – 3 – 5.4 – 2.9 – 4.8

Su programa deberá transmitir estos tres arreglos a una función llamada calc_voltios(), la cual deberá cálcular los elementos en el arreglo voltios como el producto de los elementos correspondientes en los arreglos corriente y resistencia (voltios[1] = corriente [1] * resistencia[1]).

Después que calc_voltios() ha puesto valores en el arreglo voltios, los valores en el arreglo deberán desplegarse desde adentro de main.

Codificación:

```
#include <iostream>
using namespace std;
void calc_voltios(double[ ],double[ ],double[ ]);
const int numel=10;

int main( )
{
 double corriente [numel]= {10.62,14.89,13.21,16.55,18.62,9.47,6.58,18.32,12.15,3.98};
 double voltios [numel], resistencia [numel];
 int i;
 for ( i=0 ; i<numel ; i++ )
 { cout<<"Ingrese el elemento"<<i+1<< "del arreglo";
   cin>>resistencia[i];

 }
 calc_voltios(corriente, resistencia, voltios);
 for ( i=0 ; i<numel ; i++ )
 {
 cout<<"El elemento del arreglo voltios"<<i + 1<<"es "<<voltios[i]<<endl;
 }

 system("PAUSE");
 return 0;
}
```

Prototipo void, ya que no devuelve modifica el arreglo voltios directamente

Declaración e inicialización del arreglo

Codificación:

```
void calc_voltios(double corr[NUMEL],double res[NUMEL],double volts[NUMEL]);
{
  int i;
  for (  i=0 ;  i<numel  ;   i++   )
  {
    volts[i] = res[i] * corr[i];

  }
  return;
}
```

En este programa solo se crean 3 arreglos que se conocen con dos nombres cada uno, en main voltios en calc_voltios como volts y así sucesivamente. La función recibe la dirección inicial del arreglo NO UNA COPIA

```
Por favor ingrese los 10 valores de resistencia:
4  8.5  6  7.35  9  15.3  3  5.4  2.9  4.8
El elemento del arreglo voltios 1 es: 42.48
El elemento del arreglo voltios 2 es: 126.565
El elemento del arreglo voltios 3 es: 79.26
El elemento del arreglo voltios 4 es: 121.643
El elemento del arreglo voltios 5 es: 167.58
El elemento del arreglo voltios 6 es: 144.891
El elemento del arreglo voltios 7 es: 19.74
El elemento del arreglo voltios 8 es: 98.928
El elemento del arreglo voltios 9 es: 35.235
El elemento del arreglo voltios 10 es: 19.104

Presione una tecla para continuar . . .
```

La salida generada por la ejecución del programa

Laboratorio Clase 10

TAKE HOME – TAREA

Ejercicio 2

Indicar el resultado mostrado en pantalla luego de la ejecución del programa cuando se ingresa el valor (a = 12)

```
#include <iostream>
#include <iomanip>
using namespace std;
int rango(int, int, int, int []);
const int TAM = 100;
int main()
{
int i, a, lista[TAM];
cout << "Ingrese a: ";
cin >> a;
i = rango(a, 2*a, 2, lista) - 2;
while (i > 0){
lista[i] = i + 1;
i -= 3;
}
cout << lista[4];

cin.ignore(2);
return 0;
}

int rango(int a, int b, int d, int arreglo[])
{
 int largo = 0;
 for (int i = a ; i < b; i = i + d){
 arreglo[largo] = i;
 largo++;
}
return largo;
}
```

Laboratorio Clase 10

Ejercicio 2

(a = 12)

```
i = rango(a, 2*a, 2, lista) - 2;
```

A "i" asigna el valor devuelto por la funcion rango – 2.
Los parámetros enviados son: 12 – 24 – 2 – Arreglo lista

```
int rango(int a, int b, int d, int arreglo[])
{
 int largo = 0;
 for (int i = a ; i < b; i = i + d){
 arreglo[largo] = i;
 largo++;
 }
 return largo;
}
```

Argumentos:

a= 12
b= 24
d = 2
arreglo = lista

Laboratorio Clase 10

Ejercicio 2

(a = 12)

```
int rango(int a, int b, int d, int arreglo[])
{
 int largo = 0;
 for (int i = a ; i < b; i = i + d){
 arreglo[largo] = i;
 largo++;
 }
 return largo;
}
```

for (i= 12; i < 24; i=i+2)
{arreglo[largo] = i;
largo++;}

return largo;

Argumentos:

a= 12
b= 24
d = 2
arreglo = lista

Valores de i:

12
14
16
18
20
22

Laboratorio Clase 10

```
int rango(int a, int b, int d, int arreglo[])
{
 int largo = 0;
 for (int i = a ; i < b; i = i + d){
 arreglo[largo] = i;
 largo++;
 }
 return largo;
}
```

Argumentos:

a= 12
b= 24
d = 2
arreglo = lista

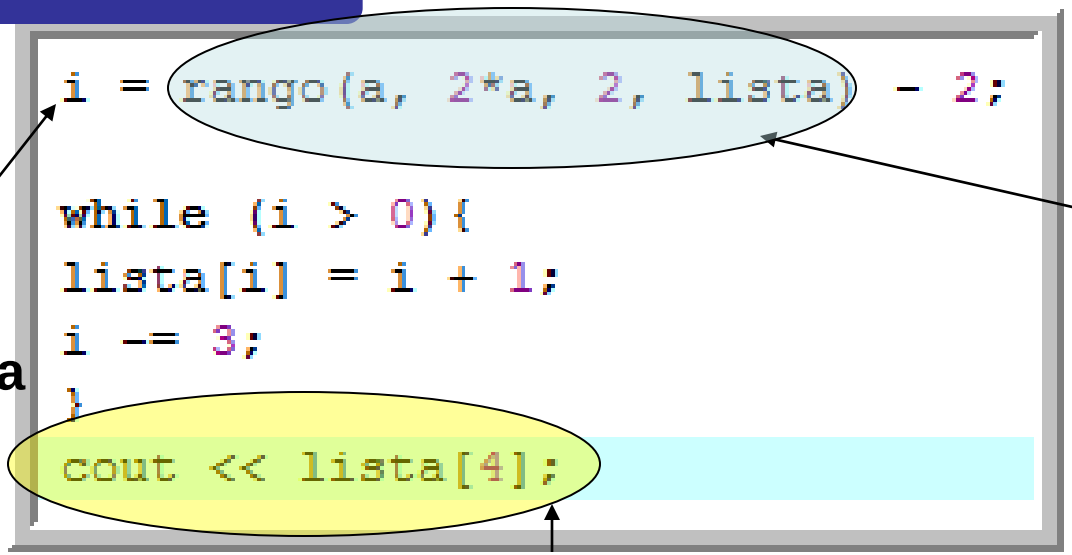
arreglo[0] = 12 **incrementa largo = 1** **incremento i = 14**
arreglo[1] = 14 **incrementa largo = 2** **incremento i = 16**
arreglo[2] = 16 **incrementa largo = 3** **incremento i = 18**
arreglo[3] = 18 **incrementa largo = 4** **incremento i = 20**
arreglo[4] = 20 **incrementa largo = 5** **incremento i = 22**
arreglo[5] = 22 **incrementa largo = 6** **incremento i = 24**

Valor devuelto por la función largo = 6

Valores de i:

12
14
16
18
20
22

Laboratorio Clase 10



Valor retornado por la función
6

En i se almacena
4 (6 – 2)

La respuesta es: 5.

El programa imprime el elemento de subíndice 4 del arreglo.

lista[4]=5

lista[1]=2

Mientras 4>0
{
lista[4] = 4 + 1;

i = 4 – 3;
}

Mientras 1>0
{
lista[1] = 1 + 1;

i = 1 – 3;
}

Laboratorio Clase 10

TAKE HOME

Ejercicio 3

```
#include <iostream>
#include <iomanip>
using namespace std;
int a, b;
int Uno(int);
int Dos(void);
int main()
{
 int c, d, e, f;
 cout << "Ingrese dos numeros enteros: " << endl;
 cout << "a = "; cin >> a; cout << endl;
 cout << "b = "; cin >> b; cout << endl;
 c = Uno(a);
 d = Uno(b);
 e = Dos();
 f = Dos();
 cout << c << d << e << f;
 cin.ignore(2);
 return 0;
}
```

```
int Uno(int p)
{
 char m = 'a', n = 'A';
 int r;
 r = int( m ) + int( n ) + p;
 return r;
}

int Dos(void)
{
 static int i = 1;
 int p;
 p = (a + b) * i;
 i++;
 return p;
}
```

Indicar el resultado mostrado en pantalla luego de la ejecución cuando se ingresan los valores (a = 2) y (b = 3)

Laboratorio Clase 10

TAKE HOME

Ejercicio 3

```
c = Uno(a);
```

A c, asignar el resultado de la función Uno con parámetro a = 2.

```
int Uno(int p)
{
 char m = 'a', n = 'A';
 int r;
 r = int( m ) + int( n ) + p;
 return r;
}
```

Carácteres imprimibles

Dec	Hex	Car.	Dec	Hex	Car.	Dec	Hex	Car.
32	20	Espacio	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d

r = 97 + 65 + 2;
devuelve r;

En c se almacenó 164.

Laboratorio Clase 10

TAKE HOME

Ejercicio 3

```
d = Uno(b);
```

A d, asignar el resultado de la función Uno con parámetro b = 3.

```
int Uno(int p)
{
 char m = 'a', n = 'A';
 int r;
 r = int( m ) + int( n ) + p;
 return r;
}
```

Carácteres imprimibles								
Dec	Hex	Car.	Dec	Hex	Car.	Dec	Hex	Car.
32	20	Espacio	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d

r = 97 + 65 + 3;
devuelve r;

En d se almacenó 165.

TAKE HOME
Ejercicio 3

```
e = Dos();
```

A e, asignar el resultado de la función Dos sin parámetros.

```
int Dos(void)
{
 static int i = 1;
 int p;
 p = (a + b) * i;
 i++;
 return p;
}
```

Variable estática que mantiene su resultado = 1. (a y b vbles. globales!)

p = (2 + 3) * 1;
devuelve 5;

En e se almacena 5.

TAKE HOME
Ejercicio 3

```
f = Dos();
```

A f, asignar el resultado de la función Dos sin parámetros.

```
int Dos(void)
{
 static int i = 1;
 int p;
 p = (a + b) * i;
 i++;
 return p;
}
```

Variable estática que mantiene su resultado = 2!. (por el incremento)

p = (2 + 3) * 2;
devuelve 10;

En f se almacena 10.

TAKE HOME
Ejercicio 3

El resultado del ejercicio es:

```
cout << c << d << e << f <<endl;
```

Tener cuidado con la impresión de espacios en blanco!!!

164165510

La ejecución del programa será:

```
Ingrese dos numeros enteros:
a = 2

b = 3

164165510
```

- Ejercicio: 1

- Desarrolle un programa que declare en la función MAIN, un arreglo bidimensional de INTs de 3 filas y 3 columnas.
- Desarrolle una función, con acceso a dicho arreglo

void cargar(int [tam][tam],int);

en la cual se le ingresen por teclado los elementos al arreglo mencionado. Los valores del arreglo deben ser todos pares, si se ingresa un valor impar se debe indicar por pantalla el error y solicitar el reingreso. Una vez ingresado completo se debe imprimir el arreglo desde el programa principal

- Declare otra función, con acceso al arreglo, que devuelva al programa principal la suma total de todos sus elementos. La misma debe imprimirse desde el main

- Ej: 2 4 6
  8 10 12
  16 18 20

El total es: 96

- Ejercicio: 1

- Desarrolle un programa que declare en la función MAIN, un arreglo bidimensional de INTs de 3 filas y 3 columnas.
- Desarrolle una función, con acceso a dicho arreglo

void cargar(int [tam][tam],int);

en la cual se le ingresen por teclado los elementos al arreglo mencionado. Los valores del arreglo deben ser todos pares, si se ingresa un valor impar se debe indicar por pantalla el error y solicitar el reingreso. Una vez ingresado completo se debe imprimir el arreglo desde el programa principal

- Declare otra función, con acceso al arreglo que imprima por pantalla la suma por columnas de los elementos de la matriz. Estos deben almacenarse en un arreglo.

- Ej: 2 4 6
  8 10 12
  16 18 20

Debera imprimir los siguientes totales: 26 32 38

Ejercicio: 2

- Desarrolle un programa que declare en la función MAIN, un arreglo bidimensional de DOUBLES de 2 filas y 2 columnas.
- Desarrolle una función, con acceso a dicho arreglo, en la cual se le ingresen por teclado los elementos al arreglo mencionado. Los valores del arreglo deben ser todos positivos o 0, si se ingresa un valor negativo se debe indicar por pantalla el error y solicitar el reingreso. Una vez ingresado completo se debe imprimir el arreglo.
- Desarrolle otra función que reciba los valores de los elementos de la diagonal principal del arreglo bidimensional, calcule el producto de los mismos e imprima por pantalla este resultado (pasaje por valor).
- Declare otra función, con acceso al arreglo que imprima por pantalla la suma de todos sus elementos y el promedio de los mismo.
- Modifique su programa para que el arreglo pueda ser N x N

Ejercicio: 3

- Desarrolle un programa que declare en la función MAIN, un arreglo bidimensional de DOUBLES de 3 filas y 3 columnas.

- Desarrolle una función, con acceso a dicho arreglo, en la cual se le ingresen por teclado los elementos al arreglo mencionado.

- Declare en el main, otro arreglo de igual dimension (3x3) para almacenar la misma matriz pero de la parte entera de cada elemento de la matriz original. Desarrolle otra función que reciba los valores de los elementos de la matriz original por valor y devuelva la parte entera. Utilizar de manera OBLIGADA el siguiente prototipo:

int cambio (double);

- Analize como resolveria el mismo problema pero si el argumento de la funcion fuera: void cambio (double [ ], int[ ])

Ejercicio 4:

- Escriba un programa en C++ que le solicite el ingreso mediante teclado de una **matriz de 4 por 4**, a través de una función realice la **suma de los elementos de cada fila de la misma**, guarde los resultados **en un arreglo**.
- Através de otra función se le solicita que realice la multiplicación de los elementos de la **diagonal principal** siempre y cuando estos sean distintos de 0, caso contrario imprimir mensaje:

  “Existe elemento NULO en diagonal principal”.
- Una vez ejecutadas ambas funciones, desde main se deberá desplegar en pantalla los resultados.
- Pruebe el funcionamiento de su programa con las dos siguientes matrices:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ -1 & -2 & 3 & -4 \\ -5 & -6 & -7 & 8 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ejercicios para tener en cuenta

Recomendaciones de ejercicios para ver de la bibliografía:

Sección 6.6 Ej 3 .
Sección 11.1 Ej 5.
Sección 11.2 Ej 3.
Sección 11.3 Ej 3.
Sección 11.3 Ej 5.
Sección 11.4 Ej 2.

Mas repasar todos lo vistos en clase y los prácticos de laboratorio.