



MATERIA: Programación I
GUIA DE LABORATORIO #: 20, 21
Fecha:

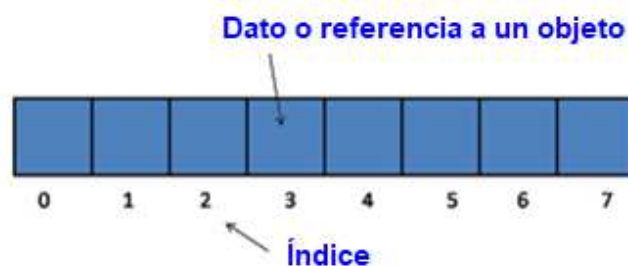
COMPETENCIAS:

Descripción de los arreglos unidimensionales y la resolución de problemas de programación que involucren arreglos.

TEORÍA ASOCIADA:

1. INTRODUCCIÓN

Un arreglo unidimensional es una estructura de datos básica que se asocia con un único identificador y puede contener cero, uno o muchas celdas contiguas, las que se numeran de forma consecutiva a partir de 0 y donde se almacenan datos del mismo tipo, a los que se puede acceder de forma individual, para lectura o escritura, como si se tratase de variables simples, mediante la combinación del identificador del arreglo y el índice de cada celda.



En programación es común denominar vectores a los arreglos unidimensionales, por la similitud de éstos con los vectores usados en el área de la matemática.

2. DECLARACIÓN E INICIALIZACIÓN DE ARREGLOS

Para declarar un arreglo en C++, se especifica el tipo de datos de los elementos que contendrá el mismo y se añade el tamaño del arreglo o número de celdas encerrado entre corchetes. Por ejemplo, para declarar un arreglo de enteros de tamaño 5, se escribe:

```
int vec[5]; //el arreglo tiene 5 celdas y los valores contenidos pueden ser arbitrarios.
```

Para inicializar un arreglo en el momento de la declaración, se usa las siguientes notaciones:

```
int vec[5] = {10, 20, 30, 40, 50}; // asigna valores iniciales a todas las celdas.
```

```
int vec[5] = {10, 20, 30}; // asigna valores iniciales a las tres primeras celdas y 0 al resto
```

```
int vec[5] = {}; // asigna valores iniciales iguales a 0 a todas las celdas.
```



3. ACCESO A LOS ELEMENTOS DEL ARREGLO

El acceso al contenido de arreglo se realiza mediante el nombre del arreglo y el índice, y se puede gestionar su contenido (leer o escribir) tal como si se tratase de una variable simple, por ejemplo:

```
int dato = vec[2];
```

```
vec[3] = 749;
```

Además, el índice puede ser una variable, con lo que puede accederse a cualquier celda, tan solo variando el valor de esa variable; por ejemplo, para imprimir en una línea todo el contenido del arreglo anterior, el código es:

```
for(int i=0; i<5; i++)  
    cout<<vec[i]<<" ";  
cout<<endl;
```

EJERCICIOS RESUELTOS:

Problema1:

Escribir un programa que lea un número natural: n ($0 < n < 15$), declare un arreglo de cadenas, luego lea los nombres de n estudiantes de la materia de Programación I, los cartue en el arreglo y posteriormente los imprima en pantalla.

EJEMPLO DE ENTRADA

3

Ana Pedro Juan

EJEMPLO DE SALIDA

Ana Pedro Juan

```
#include<iostream>  
using namespace std;  
int main()  
{  
    int n;  
    cin>>n;  
    string estudiantes[n];  
    for(int i=0;i<n;i++)  
        cin>>estudiantes[i];  
    for(int i=0;i<n;i++)  
        cout<<estudiantes[i]<<" ";  
    cout<<endl;  
    return 0;  
}
```

Problema2:



Escribir un programa que lea un número natural: n ($0 < n < 10$), luego lea n notas de estudiantes, correspondientes a las prácticas de laboratorio, por último, imprimir las dos notas más altas a partir de la segunda nota más alta.

EJEMPLO DE ENTRADA

5

58 29 74 69 70

EJEMPLO DE SALIDA

70 74

```
#include<iostream>
using namespace std;
int main()
{
    int n, mayor1=0, mayor2=0;
    cin>>n;
    int nota[n]={};
    for(int i=0;i<n;i++){
        cin>>nota[i];
        if(nota[i]>mayor1){
            mayor2=mayor1;
            mayor1=nota[i];
        }else if(nota[i]>mayor2)
            mayor2=nota[i];
    }
    cout<<mayor2<<" "<<mayor1<<endl;
    return 0;
}
```

Problema3:

Escribir un programa que cargue un vector de tamaño 30 con números aleatorios de un solo dígito, los imprima en una línea y luego imprima si se encuentran en ese arreglo todos los dígitos desde el 0 hasta el 9 sin importar el orden de los mismos.

EJEMPLO DE ENTRADA

EJEMPLO DE SALIDA

```
#include<iostream>
#include<math.h>

using namespace std;
int main()
```



```
{  
    srand(time(NULL));  
    int v[25]={};  
    for(int i=0;i<25;i++)  
        v[i]=rand()%10;  
    for(int i=0;i<25;i++)  
        cout<<v[i]<<" ";  
    cout<<endl;  
    bool digitos[10]={};  
    for(int i=0;i<25;i++)  
        digitos[v[i]]=1;  
    bool todos=true;  
    for(int i=0;i<10;i++)  
        if(digitos[i]==0)  
            todos=false;  
    if(todos)  
        cout<<"SI"<<endl;  
    else  
        cout<<"NO"<<endl;  
    return 0;  
}
```

EJERCICIOS PARA RESOLVER EN CLASES

Problema4:

Escribir un programa que cargue un vector de tamaño 10 con números aleatorios positivos y negativos de un solo dígito, luego imprima si existe igual, menor o mayor cantidad de números positivos que negativos.

Suponiendo que, al ejecutar el programa, el arreglo se carga con los valores que siguen:

-2	4	8	-9	0	4	-6	-2	5	0
----	---	---	----	---	---	----	----	---	---

Entonces la salida del programa será:

EJEMPLO DE ENTRADA

EJEMPLO DE SALIDA:

Existe mayor cantidad de números positivos que negativos.

Problema 5:

Escriba un programa que cargue dos vectores de tamaño 10 con números aleatorios comprendidos entre 10 y 100 inclusive, luego halle el promedio de las celdas de igual índice en un tercer vector e imprima los tres arreglos en tres líneas diferentes.

Suponiendo que los dos arreglos se cargaron con los siguientes números aleatorios:

80	45	58	86	45	49	55	67	75	62
----	----	----	----	----	----	----	----	----	----

74	63	87	90	78	74	56	54	80	90
----	----	----	----	----	----	----	----	----	----

La salida del programa será:



EJEMPLO DE ENTRADA

EJEMPLO DE SALIDA

77 54 72.5 88 61.5 61.5 55.5 60.5 77.5 76

EJERCICIOS PARA RESOLVER EXTRACLASE:

Problema6:

Escribir un programa que lea un número natural: n ($0 < n < 100$), declare un arreglo de tamaño: n , cargue en él todos los números desde el 1 hasta n inclusive en posiciones aleatorias, luego imprima el arreglo en una sola línea.

EJEMPLO DE ENTRADA

12

EJEMPLO DE SALIDA

4 7 10 2 9 8 1 6 11 3 5 12

Problema7:

Escribir un programa que permita guardar los nombres y las edades de cada uno de los estudiantes de la materia de Programación I. El proceso de lectura de datos terminará cuando se introduzca como nombre un asterisco (*). Al finalizar se mostrará los siguientes datos:

- Todos los estudiantes mayores de 20 años y menores de 30 años.
- Los estudiantes adultos (los que tienen una edad igual o mayor a 30 años)

EJEMPLO DE ENTRADA

José Antonio Miranda

18

Ana María Pacheco

20

Juan Palala

17

Mildred Tapia Ontiveros

32

René Palala

19

EJEMPLO DE SALIDA

MAYORES:

Ana María Pacheco

20

ADULTOS

Mildred Tapia Ontiveros

32

Problema8:



Escribir un programa que lea 5 números por teclado, los almacene en un vector y los ordene de forma ascendente. Al finalizar mostrara por pantalla el nuevo vector con los números ordenados.

EJEMPLO DE ENTRADA

58 23 95 73 17

EJEMPLO DE SALIDA

17 23 58 73 95

Problema9:

Escribir un programa que permita leer dos números enteros: a, b ($0 < \{a, b\} < 100$; $a < b$), cargar en un vector los todos los números primos comprendidos entre a y b inclusive y luego imprimirlos en la pantalla.

EJEMPLO DE ENTRADA

10 20

EJEMPLO DE SALIDA

11 13 17 19