

FUNDAMENTOS DE PROGRAMACIÓN

Inicio Clase 08

Profesor: Carlos Díaz

Clase 08: Arreglos unidimensionales

- Array unidimensional
- Inicializar un array

Array unidimensional

- Todas las variables que se han usado hasta ahora solo podían almacenar un solo dato a la vez.

```
int x;
```

```
x=7;
```

```
x=8;
```

- Un array es un conjunto de datos del mismo tipo, de tal forma que se puede utilizar para almacenar varios datos utilizando un solo nombre de variable.

```
int x[3];
```

```
x[0]=7;
```

```
x[1]=8;
```

```
x[2]=9;
```

Ejercicio 1

Escriba un programa que primero lea 5 valores.

Luego imprima sus cuadrados y su suma.

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int x[5],s=0;
```

```
    for (int i=0;i<=4;i++)
```

```
    {
```

```
        cout<<"Ingrese x("<<i<<"= ";
```

```
        cin>>x[i];
```

```
    }
```

```
    cout<<"Usted ingreso:\n";
```

```
    for (int i=0;i<=4;i++)
```

```
    {
```

```
        cout<<"x("<<i<<"= "<<x[i]<<endl;
```

```
        s=s+x[i]*x[i];
```

```
    }
```

```
    cout<<s<<endl;
```

```
    return 0;
```

```
}
```

Ejercicio 2

Escriba un programa que lea:

El nombre de N estudiantes.

La nota de práctica, examen parcial y examen final.

Calcule y muestre:

- a) La nota final de cada estudiante (promedio de práctica y exámenes)
- b) El promedio del aula
- c) Los nombres de los estudiantes que aprobaron (nota final ≥ 10)

Solución

```
#include <iostream>
using namespace std;
int main()
{
    int cantidad;
    cout<<"Ingrese la cantidad de estudiantes: ";
    cin>>cantidad;
    string nombre[cantidad], apellido[cantidad];
    double notafinal[cantidad], np, ep, ef, paula=0;
    for (int i=0; i<cantidad; i++)
        //Ingreso de datos
        {
            cout<<"Nombre y apellido "<<i+1<<": ";
            cin>>nombre[i]>>apellido[i];
            cout<<"Nota practica: ";
            cin>>np;
            cout<<"Examen parcial: ";
            cin>>ep;
            cout<<"Examen final: ";
            cin>>ef;
            notafinal[i]=(np+ep+ef)/3;
        }
}
```

```

//Muestra estudiante y nota final y promedio del aula
cout<<"-----"<<endl;
cout<<"Estudiante      Nota Final"<<endl;
for (int i=0;i<cantidad;i++)
{
    cout<<nombre[i]<<" "<<apellido[i]<<" " <<notafinal[i]<<endl;
    paula=paula+notafinal[i];
}
cout<<"Promedio del aula: "<<paula/cantidad<<endl;
//Estudiantes que aprobaron
cout<<"-----"<<endl;
cout<<"Estudiantes aprobados"<<endl;
for (int i=0;i<cantidad;i++)
{
    if (notafinal[i]>=10)
        cout<<nombre[i]<<" "<<apellido[i]<<" " <<notafinal[i]<<endl;
}
return 0;
}

```

Inicializar un array

```
#include <iostream>
using namespace std;
int main()
{
    int x[5]={7,2,-5,3,9}; //Inicializa el array
    int y[5]={}; //Inicializa con ceros
    for (int i=0;i<5;i++)
        cout<<x[i]<<endl;
    cout<<"-----"<<endl;
    for (int i=0;i<5;i++)
        cout<<y[i]<<endl;
    return 0;
}
```


Ordenamiento de burbuja

elemento(1)=	65
elemento(2)=	18
elemento(3)=	32
elemento(4)=	53
elemento(5)=	20
elemento(6)=	7

n=6

i=1

						Queda
65 elemento(1)	18	18	18	18	18	18
18 elemento(2)	65 elemento(2)	32	32	32	32	32
32	32 elemento(3)	65 elemento(3)	53	53	53	53
53	53	53 elemento(4)	65 elemento(4)	20	20	20
20	20	20	20 elemento(5)	65 elemento(5)	7	7
7	7	7	7	7 elemento(6)	65	65

i=2

18 elemento(1)	18	18	18	18	18	18
32 elemento(2)	32 elemento(2)	32	32	32	32	32
53	53 elemento(3)	53 elemento(3)	20	20	20	20
20	20	20 elemento(4)	53 elemento(4)	7	7	7
7	7	7	7 elemento(5)	53 elemento(5)	53	53
65	65	65	65	65 elemento(6)	65	65

i=3

18	elemento(1)	18		18		18		18		18
32	elemento(2)	32	elemento(2)	20		20		20		20
20		20	elemento(3)	32	elemento(3)	7		7		7
7		7		7	elemento(4)	32	elemento(4)	32		32
53		53		53		53	elemento(5)	53	elemento(5)	53
65		65		65		65		65	elemento(6)	65

i=4

18	elemento(1)	18		18		18		18		18
20	elemento(2)	20	elemento(2)	7		7		7		7
7		7	elemento(3)	20	elemento(3)	20		20		20
32		32		32	elemento(4)	32	elemento(4)	32		32
53		53		53		53	elemento(5)	53	elemento(5)	53
65		65		65		65		65	elemento(6)	65

i=5

18	elemento(1)	7		7		7		7		7
7	elemento(2)	18	elemento(2)	18		18		18		18
20		20	elemento(3)	20	elemento(3)	20		20		20
32		32		32	elemento(4)	32	elemento(4)	32		32
53		53		53		53	elemento(5)	53	elemento(5)	53
65		65		65		65		65	elemento(6)	65

Algoritmo ordenamiento de burbuja

' Suponiendo que hay "n" elementos

' Suponiendo que desamos orden ASCENDENTE

Para $i \leftarrow 1$ hasta $n-1$

Para $j \leftarrow 1$ hasta $n-1$

Si $\text{elemento}(j) > \text{elemento}(j+1)$ Entonces

temp = elemento(j)

elemento(j) = elemento(j+1)

elemento(j+1) = temp

Fin de SI

Fin de Para

Fin de Para

Ejercicio 3

Utilice un arreglo unidimensional para resolver el siguiente problema. Una compañía paga a sus vendedores por comisión. Los vendedores reciben \$200 por semana más 9% de sus ventas totales de esa semana. Por ejemplo, un vendedor que acumule \$5000 en ventas en una semana, recibirá \$200 más 9% de \$5000, o un total de \$650. Escriba un programa (utilizando un arreglo de contadores) que determine cuántos vendedores recibieron salarios en cada uno de los siguientes rangos (suponga que el salario de cada vendedor se trunca a una cantidad entera):

- a) \$200-299
- b) \$300-399
- c) \$400-499
- d) \$500-599
- e) \$600-699
- f) \$700-799
- g) \$800-899
- h) \$900-999
- i) \$1000 en adelante

Ejercicio 4

(*Ordenamiento de burbuja*) En el **ordenamiento de burbuja**, los valores más pequeños van “subiendo como burbujas” gradualmente, hasta llegar a la parte superior del arreglo (es decir, hacia el primer elemento) como las burbujas de aire que se elevan en el agua, mientras que los valores más grandes se hunden en el fondo. Esta técnica realiza varias pasadas a través del arreglo. En cada pasada compara pares sucesivos de elementos. Si un par se encuentra en orden ascendente (o los valores son idénticos), el ordenamiento de burbuja deja los valores como están. Si un par se encuentra en orden descendente, el ordenamiento de burbuja intercambia sus valores en el arreglo. Escriba un programa que ordene un arreglo de 10 enteros mediante el uso del ordenamiento de burbuja.

Ejercicio 5

Use un arreglo unidimensional para resolver el siguiente problema. Recibir como entrada 20 números, cada uno de los cuales debe estar entre 10 y 100, inclusive. A medida que se lea cada número, validarlo y almacenarlo en el arreglo, sólo si no es un duplicado de un número ya leído. Después de leer todos los valores, mostrar sólo los valores únicos que el usuario introdujo. Prepárese para el “peor caso”, en el que los 20 números son diferentes. Use el arreglo más pequeño que sea posible para resolver este problema.

Ejercicio 6

Se tiene un array de enteros cualesquiera. Obten otro de forma que el contenido de cada elemento del nuevo array sea un índice que nos indique de menor a mayor los valores del array de enteros. Por ejemplo:


0	1	2	3	4
10	5	-7	0	12

Valor

0	1	2	3	4
2	3	1	0	4

Indice

Ejercicio 7

Escriba un programa para simular el tiro de dos dados. El programa debe utilizar `rand` para tirar el primer dado, y de nuevo para tirar el segundo dado. Después debe calcularse la suma de los dos valores. [Nota: cada dado puede mostrar un valor entero del 1 al 6, por lo que la suma de los valores variará del 2 al 12, siendo 7 la suma más frecuente, mientras que 2 y 12 serán las sumas menos frecuentes]. En la figura  se muestran las 36 posibles combinaciones de los dos dados. Su programa debe tirar los dados 36,000 veces. Utilice un arreglo unidimensional para registrar el número de veces que aparezca cada una de las posibles sumas. Imprima los resultados en formato tabular. Determine además si los totales son razonables (es decir, hay seis formas de tirar un 7, por lo que aproximadamente una sexta parte de los tiros deben ser 7).

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

Ejercicio 8

La desviación estándar de una lista de números es una medida de qué tanto los números difieren del promedio. Si la desviación estándar es pequeña, los números están agrupados cerca del promedio. Si la desviación estándar es grande, los números están dispersos lejos del promedio. La desviación estándar, S , de una lista de N números x_i se define así:

$$S = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}}$$

donde \bar{x} es la media de los N números x_1, x_2, \dots . Defina un arreglo de números que devuelva la desviación.

INFORMÁTICA

C++

Fin Clase 07

Profesor: Carlos Díaz