

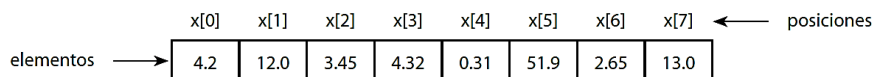
Clase 06: Arrays unidimensionales

- Arreglos unidimensionales
- Inicialización de arreglos unidimensionales
- Tamaño inicial del array
- Arreglo de caracteres
- Ejemplos

Arreglos unidimensionales

Lenguaje C

```
float x[8];
```



Lenguaje C

```
a = x[0] + x[1] + x[2];
printf("%f", a);
```

Lenguaje C

```
int x[3] = 18;
```

Inicialización de arreglos unidimensionales

Lenguaje C	Lenguaje C
<pre> tipo_dato identif[tam_arreglo]={valores}; int lista [5] = {10,17,8,4,9}; </pre>	<pre> #include <stdio.h> #include <conio.h> main() { int lista[10], i; for (i=0; i<=9;i++) { printf("Dame el elemento"); scanf("%d",&lista[i]); } printf("Elementos de la lista\n"); for (i=0; i<=9;i++) printf("%d\n",lista[i]); getch(); return 0; } </pre>

Tamaño inicial del array

Lenguaje C (Turbo C, DEV-CPP y Code::Blocks)	DEV-CPP y Code::Blocks
<pre> #define MAX 30 #include <stdio.h> #include <conio.h> main () { int lista[MAX], i,n; printf ("Dame el tamaño del arreglo: "); scanf ("%d",&n); for (i=0; i<n; i++) { printf("Dame el elemento %d ",i); scanf("%d",&lista[i]); } printf("Elementos de la lista\n"); for (i=0; i<n;i++) printf("%d\n",lista[i]); getch(); return 0; } </pre>	<pre> #include <stdio.h> #include <conio.h> main () { int i,n; printf ("Dame el tamaño del arreglo: "); scanf ("%d",&n); int lista[n]; for (i=0; i<n;i++) { printf("Dame el elemento %d ",i); scanf("%d",&lista[i]); } printf("Elementos de la lista\n"); for (i=0; i<n;i++) printf("%d\n",lista[i]); getch(); return 0; } </pre>

Arreglo de caracteres

Los arreglos son necesarios para la implementación de cadenas de caracteres. Una cadena de texto es un conjunto de caracteres. Hay que recordar que en el lenguaje C no existe el tipo de dato cadena (string) como en otros lenguajes de programación, por lo que se utiliza un arreglo de caracteres, para poder almacenar una cadena:

Lenguaje C

```
char cad[ ]="Lenguaje";
```

Una cadena de caracteres es un arreglo de caracteres que contiene al final el carácter nulo (\0); por esta razón es necesario que al declarar los arreglos éstos sean de un carácter más que la cadena más grande. El compilador inserta automáticamente un carácter nulo al final de la cadena, de modo que la secuencia real sería:

Lenguaje C

```
char cad[9]="Lenguaje";
```

La cadena quedaría almacenada como sigue:

Contenido	L	e	n	g	u	a	j	e	\0
Elemento del arreglo	cad[0]	cad[1]	cad[2]	cad[3]	cad[4]	cad[5]	cad[6]	cad[7]	cad[8]

Arreglo de caracteres

Una opción para almacenar una cadena de caracteres es el uso de la palabra reservada *scanf* (variable) pero, si queremos almacenar una cadena con espacios en blanco no lo podemos hacer con ella, sino que debemos utilizar la palabra reservada *gets*, que se encuentra dentro de la librería *string.h*; *gets* sólo se utiliza para leer cadenas de caracteres y *scanf* para leer cualquier tipo de variable, de preferencia de tipo numérico.

gets

Introduce una cadena de caracteres del teclado hasta que se encuentra un carácter '\n' (nueva línea), dicho carácter no se añade a la cadena. Se agrega al final del arreglo un carácter de terminación NULL.

Sintaxis: gets(variable_cadena);

scanf

Lee una cadena. El argumento correspondiente es un apuntador a un arreglo del tipo char, que es lo suficiente extenso para contener la cadena y un carácter de terminación NULL.

Sintaxis: scanf ("%s", &variable_cadena);

Cuando inicializamos una cadena de caracteres no es necesario definir el tamaño, el compilador siempre añade un carácter nulo al final.

```
char cadena[ ] = "Programación" //cadena de 13 caracteres
```

Ejemplo

Escriba un programa que determine si una palabra es palíndromo o no lo es. (Un palabra es palíndromo si se lee de igual manera al derecho y al revés).

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
main()
{
    char palabra[80];
    int x,y,z;
    printf("\t Programa que determina si una palabra es palindromo");
    printf("\n\nEscribe una palabra: ");
    scanf(" %s",&palabra);
    x=strlen(palabra);
    x=x-1;
    for(y=0,z=x;y<=x/2;y++,z--)
        if(palabra[y]==palabra[z])
            printf("Es palindromo");
        else
            printf("No es palindromo");
    getch();
}
```

Ejercicio 1. Inicialice un arreglo de 10 elementos, realice e imprima la sumatoria.

Lenguaje C

```
#include <stdio.h>
#include <conio.h>
main()
{
    int x[10]={6,7,9,2,4,7,4,3,2,9};
    int i,s=0;
    for (i=0;i<10;i++)
        s+=x[i];
    printf("La suma es %d",s);
    getch(); return 0;
}
```

Ejercicio 2. Almacene la edad de todos los alumnos de un salón de clases y posteriormente preguntar e imprimir la edad de alguno de ellos.

```
Lenguaje C
#include <stdio.h>
#include <conio.h>
main()
{
    int alumn[45], total_a,i;
    printf("Total de alumnos:\n");
    scanf("%d",&total_a);
    for( i=0; i< total_a; i++)
    {
        printf("Dame la edad del alumno: %d\n",i+1);
        scanf("%d", &alumn[i]);
    }
    printf("¿De qué alumno quieres su edad?");
    scanf("%d",&i);
    printf("La edad es: %d", alumn[i-1]);
    getch(); return 0;
}
```

Ejercicio 3. Calcule el promedio de las calificaciones de las materias del semestre anterior, utilizando un arreglo para almacenar todas las calificaciones y el promedio guárdelo en la siguiente posición después de la última calificación:

```
Lenguaje C
#include <stdio.h>
#include <conio.h>
main()
{
    float cal[10],prom=0;
    int i,mat;
    printf("¿Cuántas materias llevas (max 9)?\n");
    scanf("%d",&mat);
    for (i=0;i<mat;i++)
    {
        printf("Calificación de la %d materia: ",i+1);
        scanf("%f",&cal[i]);
        prom=prom+cal[i];
    }
    cal[mat]=prom/mat;
    printf("El promedio es %4.2f",cal[mat]);
    getch();
    return 0;
}
```

Ejercicio 4. Almacene en un arreglo, n elementos de tipo entero, calcule el cuadrado de cada uno de los elementos y almacénelo en un segundo arreglo. Imprima ambos vectores.

Lenguaje C

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{
    int x[10], i, n;
    float y[10];
    printf("Dame el tamaño del arreglo:");
    scanf("%d", &n);
    for (i=0; i<n; i++)
    {
        printf("Dame un número: ");
        scanf("%d", &x[i]);
        y[i] = pow(x[i], 2);
    }
    printf("Los cuadrados son:\n");
    for(i=0; i<n; i++)
        printf("%d, %.0f \n", x[i], y[i]);
    getch(); return 0;
}
```

Ejercicio 5. Calcule la desviación estándar S de cinco números.

Lenguaje C

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{
    int num[5], i;
    float s=0.0, m=0.0;
    printf("introduce los 5 números \n");
    for (i=0; i<5; i++)
    {
        scanf("%d", &num[i]);
        m+=num[i];
    }
    m=m/5;
    for (i=0; i<5; i++)
        s+=pow((m-num[i]), 2);
    s=sqrt(s/5);
    printf("La media es: %.2f ", m);
    printf("La desviación estándar es %.2f", s);
    getch();
    return 0;
}
```

$$S = \sqrt{\frac{\sum_{i=0}^{i=4} (m - \text{num})^2}{5}}$$

Ejercicio 6. Almacene en un vector, n elementos (máximo 30) e inserte un elemento ni en la posición pos , recorriendo los siguientes elementos una posición a la derecha.

Lenguaje C

```
#include <stdio.h>
#include <conio.h>
#define p printf
#define s scanf
#define MAX 30
main()
{
    int lista[MAX],i,ni,n,pos;
    p("Dame el Tamaño del Arreglo ");
    s("%i",&n);
    p ("Dame el No. a Insertar ");
    s("%i",&ni);
    p("Dame la Posición ");
    s("%i",&pos);
    for (i=0;i<n;i++)
    {
        p ("Dame el valor cuya posición es %d ",i);
        scanf("%d",&lista[i]);
    }
    for (i=n;i>pos;i--)
        lista[i]=lista[i-1];
    lista[pos]=ni;
    for (i=0;i<=n;i++)
        p ("%d ",lista[i]);
    getch();
}
```

Ejercicio 7. Almacene en un arreglo a , un conjunto n de elementos de tipo entero (max 15), almacene en el arreglo b los elementos del arreglo a de forma invertida.

Lenguaje C

```
#include <stdio.h>
#include <conio.h>
main()
{
    int a[15],b[15],n, i,j;
    printf ( "Total de números");
    scanf ("%d",&n );
    for (i=0; i<n; i++)
        scanf ("%d",&a[i]);
    for (i=0, j=n-1; i<n, j>=0; i++, j--)
        b[i]=a[j];
    for (i=0; i<n ; i++)
        printf ("%d \n",a[ i ] );
    for (i=0; i<n ; i++)
        printf ("%d \n",b[ i ] );
    getch();
    return 0;
}
```

Arreglo $a = 2, 4, 6, 8, 10$ almacena el usuario

Arreglo $b = 10, 8, 6, 4, 2$ lo realiza el programa