



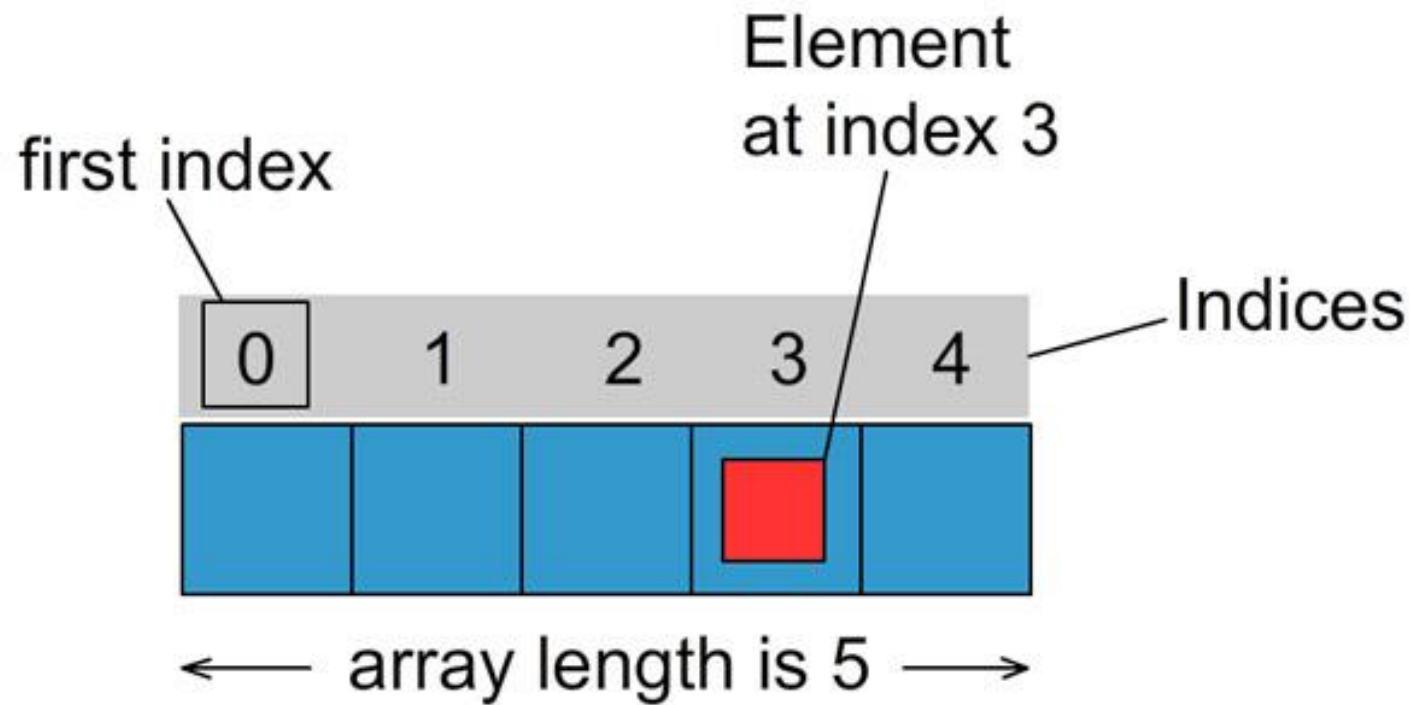
Grado en Estadística Aplicada

Programación II

Arrays Unidimensionales

CONCEPTO

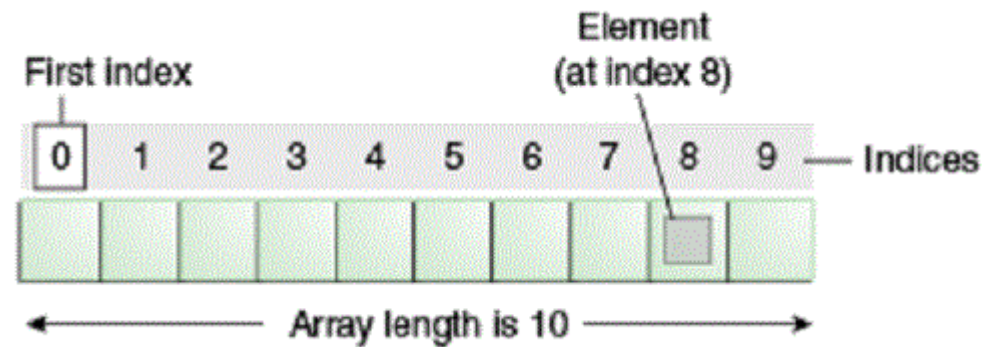
Colección de elementos del mismo tipo.





CARACTERISTICAS

- Todos los elementos son del mismo tipo
- A cada dato almacenado se le denomina *elemento del array*
- Cada elemento del array está numerado, y a éste número se le denomina *índice*.
- Los elementos del array se numeran consecutivamente comenzando con 0
- Los índices, permiten localizar cada elemento del array. `letras[0]` es el elemento que está en la posición 0.
- Los elementos almacenados en el array pueden ser de cualquier tipo: simples o estructurados





DECLARACIÓN

main.cpp

```
1  #include <iostream>
2
3  int main(int argc, char** argv) {
4      /*Programa ejemplo array unidimensional*/
5
6      const int MAX = 10; //numero de elementos
7      typedef int tArrayContador[MAX]; //definimos tipo array de MAX elementos
8
9      tArrayContador cont; //declaramos variable del tipo de array definido
10
11
12      return 0;
13  }
14
15
```



INICIALIZAR



```
1  #include <iostream>
2  using namespace std;
3
4  int main(int argc, char** argv) {
5      /*Programa ejemplo array unidimensional*/
6
7      const int MAX = 10; //numero de elementos
8      typedef int tArrayContador[MAX]; //definimos tipo array de MAX elementos
9
10     tArrayContador cont;
11     .....
12     cont = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
13
14     system("PAUSE");
15     return 0;
16 }
```

INICIALIZAR

```

1  #include <iostream>
2  using namespace std;
3
4  int main(int argc, char** argv) {
5      /*Programa ejemplo array unidimensional*/
6
7      const int MAX = 10; //numero de elementos
8      typedef int tArrayContador[MAX]; //definimos tipo array de MAX elementos
9
10     tArrayContador cont = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
11
12     system("PAUSE");
13     return 0;
14 }

```



Podemos asignar valor a todos los elementos del array en el momento de declarar la variable de tipo array.

Se asignan los valores por su orden:

cont[0] cont[1] cont[2] cont[3] cont[4] ... cont[9]
 ↑ ↑ ↑ ↑ ↑ ... ↑
 1º 2º 3º 4º 5º ... 10º



```

1  #include <iostream>
2  using namespace std;
3
4  int main(int argc, char** argv) {
5      /*Programa ejemplo array unidimensional*/
6
7      const int MAX = 10; //numero de elementos
8      typedef int tArrayContador[MAX]; //definimos tipo array de MAX elementos
9
10     tArrayContador cont;
11     .....
12     cont = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
13
14     system("PAUSE");
15     return 0;
16 }

```

Esto provoca error.



ASIGNAR/LEER VALORES

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

int main(int argc, char** argv) {
    /*Programa ejemplo array unidimensional*/
    const int MAX = 10; //numero de elementos
    typedef int tArrayContador[MAX]; //definimos tipo array de MAX elementos

    tArrayContador cont = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

    cont[0] = 1;
    cont[1] = 2;
    cont[2] = 3;
    cont[3] = 4;
    cont[4] = 5;
    cont[5] = 6;
    cont[6] = 7;
    cont[7] = 8;
    cont[8] = 9;
    cont[9] = 10;

    cout << cont[0] << "\n";
    cout << cont[1] << "\n";
    cout << cont[2] << "\n";
    cout << cont[3] << "\n";
    cout << cont[4] << "\n";
    cout << cont[5] << "\n";
    cout << cont[6] << "\n";
    cout << cont[7] << "\n";
    cout << cont[8] << "\n";
    cout << cont[9] << "\n";

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



RECORRER ARRAYS

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

int main(int argc, char** argv) {
    /*Programa ejemplo array unidimensional*/
    const int MAX = 10; //numero de elementos
    typedef int tArrayContador[MAX]; //definimos tipo array de MAX elementos

    tArrayContador a;

    int numero;

    cout << "Introduce " << MAX << " numeros:\n";
    for(int i=0;i<MAX;i++)
    {
        cin >> numero; //Se lee el numero
        a[i] = numero; //guardo el numero en la posicion indice del array
    }

    for(int i=0;i<MAX;i++)
    {
        cout << "En la posicion " << i << " esta el numero " << a[i] << "\n";
    }

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

Introduce 10 numeros:

1
2
5
65
32
3
0
84
4
7

En la posicion 0 esta el numero 1
En la posicion 1 esta el numero 2
En la posicion 2 esta el numero 5
En la posicion 3 esta el numero 65
En la posicion 4 esta el numero 32
En la posicion 5 esta el numero 3
En la posicion 6 esta el numero 0
En la posicion 7 esta el numero 84
En la posicion 8 esta el numero 4
En la posicion 9 esta el numero 7
Presione una tecla para continuar . . .



EJERCICIOS

1. Realiza un programa que guarde en un array las notas de 10 alumnos, las notas son números enteros entre 1 y 10.
2. Una vez introducida las notas mostraremos solo aquellas que son mayores o iguales que 5
3. Decir cual es la mayor nota introducida
4. Realiza un programa que recoja 9 alturas de personas en centímetros, se guardaran en un array. Cuando finalice nos dirá la media de la altura de estas personas
5. Mostrar las alturas introducidas en el ejercicio anterior en el orden inverso en el que se introdujeron
6. Guardar las alturas en orden inverso en un segundo array y mostrarlo por pantalla.



EJERCICIOS

1. Rellenar 2 arrays cuyo tamaño se indica por teclado. Rellenar un tercer array con la suma de los dos primeros (sumar contenido de la posición 1 del primer array con el contenido de la posición 1 del segundo array....)
2. Dado un numero entre 1 y 10 rellenar un array de 10 elementos con la tabla de multiplicar del numero introducido
3. Dado un array de tamaño X, obtener la suma de todos sus elementos, conocer cual es mayor, conocer cual es el menor y la media.



LABORATORIO AHORCADO MEJORADO

Se pretende implementar mediante C++ y el uso de arrays el juego del ahorcado adaptado a las siguientes normas.

Se pide una palabra de máximo 15 letras.

Se crea un array de 8 elementos todos con valor 0.

Comienza el juego:

Se introduce 8 letras:

si la palabra contiene esa letra nos dice en que posición esta dentro de la palabra.

si no contiene la letra ese intento en el array se marca con valor 1.

Una vez introducidas las 8 letras:

Si en el array están todos los elementos con valor 1 se muestra una mensaje indicando que se ha perdido la partida

En caso contrario nos pide introducir la palabra, si es la misma se gana.

```
//PISTA:  
string palabra="melocoton";  
  
for (int i=0;i<palabra.length();i++)  
{  
    cout << "Posicion " << i << "tiene la letra " << palabra[i] << "\n";  
}
```

```
Posicion 0tiene la letra m  
Posicion 1tiene la letra e  
Posicion 2tiene la letra l  
Posicion 3tiene la letra o  
Posicion 4tiene la letra c  
Posicion 5tiene la letra o  
Posicion 6tiene la letra t  
Posicion 7tiene la letra o  
Posicion 8tiene la letra n
```



LABORATORIO AHORCADO MEJORADO

Avanzado: mostrar el ahorcado.

```
Funcion MostrarAhorcado(fallos)
  Escribir "";
  Escribir "La horca!!!";
  Escribir "";
  Segun fallos Hacer
    0:
      Escribir "";
      Escribir "";
      Escribir "";
      Escribir "";
      Escribir "";
      Escribir "";
    1:
      Escribir "";
      Escribir "";
      Escribir "";
      Escribir "";
      Escribir "";
      Escribir "_____";
    2:
      Escribir "|";
      Escribir "|";
      Escribir "|";
      Escribir "|";
      Escribir "|";
      Escribir "_____";
```

```
3:
  Escribir "-----";
  Escribir "|";
  Escribir "|";
  Escribir "|";
  Escribir "|";
  Escribir "|";
  Escribir "_____";
4:
  Escribir "-----";
  Escribir "| |";
  Escribir "| o";
  Escribir "| ";
  Escribir "| ";
  Escribir "_____";
5:
  Escribir "-----";
  Escribir "| |";
  Escribir "| o";
  Escribir "| /|";
  Escribir "| ";
  Escribir "_____";
6:
  Escribir "-----";
  Escribir "| |";
  Escribir "| o";
  Escribir "| /\\";
  Escribir "| ";
  Escribir "_____";
7:
  Escribir "-----";
  Escribir "| |";
  Escribir "| o";
  Escribir "| /\\";
  Escribir "| / ";
  Escribir "_____";
8:
  Escribir "-----";
  Escribir "| |";
  Escribir "| o";
  Escribir "| /\\";
  Escribir "| / \";
  Escribir "_____";

FinSegun
Escribir "";
FinFuncion
```