Electrónica digital II

Santiago Rúa Pérez, PhD.

18 de septiembre de 2022

ARREGLOS EN C

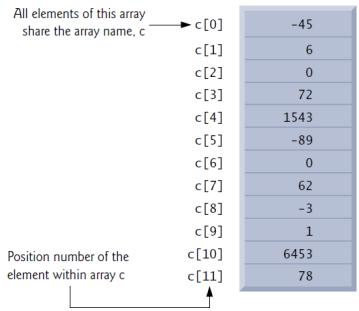
Arreglos en C

Objetivos

- Utilizar los arreglos para representar listas o tablas.
- Definir un arreglo, inicializarlo y direccionar.
- Definir constantes simbólicas.
- Pasar arreglos a funciones.
- Definir arreglos multidimensionales.
- Crear arreglos de longitud variable.

Arreglos

Es un conjunto de datos organizados de forma contigua en memoria.



Arreglos - Declaración

Para crear un vector basta con indicar el nombre del vector y el tamaño del mismo, de esta forma se le da la orden al compilador el espacio en memoria.

```
1   #include <stdio.h>
2
3   // function main begins program execution
4   int main(void)
5   {
6     int n[5]; // n is an array of five integers
7
8     // set elements of array n to 0
9     for (size_t i = 0; i < 5; ++i) {
10      n[i] = 0; // set element at location i to 0
11    }
12
13    printf("%s%13s\n", "Element", "Value");
14
15    // output contents of array n in tabular format
16    for (size_t i = 0; i < 5; ++i) {
17      printf("%7u%13d\n", i, n[i]);
18    }
19  }
20
```

Pueden inicializarse mediante lista de elementos. También se puede definir la directiva de preprocesamiento `#define`.

Inicialización - Ejemplo

```
1   #include <stdio.h>
2
3   // function main begins program execution
4   int main(void)
5   {
6     // use initializer list to initialize array n
7     int n[5] = {32, 27, 64, 18, 95};
8
9     printf("%s%13s\n", "Element", "Value");
10
11    // output contents of array in tabular format
12    for (size_t i = 0; i < 5; ++i) {
13      printf("%7u%13d\n", i, n[i]);
14    }
15  }
16
```

Inicialización variable simbólica - Ejemplo

```c
#include <stdio.h>

#define SIZE 5 // maximum size of array

// function main begins program execution
int main(void)
{
  // symbolic constant SIZE can be used to specify array size
  int s[SIZE]; // array s has SIZE elements

  for (size_t j = 0; j < SIZE ; ++j) { // set the values
    s[j] = 2 + 2 * j;
  }

  printf("%s%13s\n", "Element", "Value");

  // output contents of array s in tabular format
  for (size_t j = 0; j < SIZE ; ++j) {
    printf("%7u%13d\n", j, s[j]);
  }
}
```

Arreglos - Ejercicio

- Cuarenta estudiantes fueron encuestados sobre la calidad de comida en un restaurante es un escala de 1 a 10 siendo está última como la mejor. Ponga las 40 respuestas en un vector de enteros y encuentre el resumen de los resultados de la encuesta.
- Lance los dados 60000000 de veces y haga un resumen de los resultados.
- Haga un programa que dado un vector de tamaño n, los organice de mayor a menor.

Solución primer ejercicio

```c
#include <stdio.h>

#define SIZE 5 // maximum size of array

// function main begins program execution
int main(void)
{
  // symbolic constant SIZE can be used to specify array size
  int s[SIZE]; // array s has SIZE elements

  for (size_t j = 0; j < SIZE ; ++j) { // set the values
    s[j] = 2 + 2 * j;
  }

  printf("%s%13s\n", "Element", "Value");

  // output contents of array s in tabular format
  for (size_t j = 0; j < SIZE ; ++j) {
    printf("%7u%13d\n", j, s[j]);
  }
}

```

Cadenas

Las cadenas son un arreglo de caracteres terminados con el valor NULL. El valor NULL se puede representar mediante `'\0'`. Por ejemplo

```
1    char string1[] = "first";
2    char string1[] = {'f', 'i', 'r', 's', 't', '\0'};
3
```

Ambas cadenas son lo mismo y generan el mismo tamaño.

Para manipular cadenas, existen librerias propias dentro de C especializadas en el tratamiento de las mismas. Un ejemplo de estas funciones son: `fgets`, `puts`, `sscanf`, `sprintf`, entre otras.

Enviar arreglos a funciones

Para pasar un arreglo completo a una función se debe enviar el nombre del arreglo sin nada entre corchetes. Automáticamente C detecta que esto es un paso por referencia.

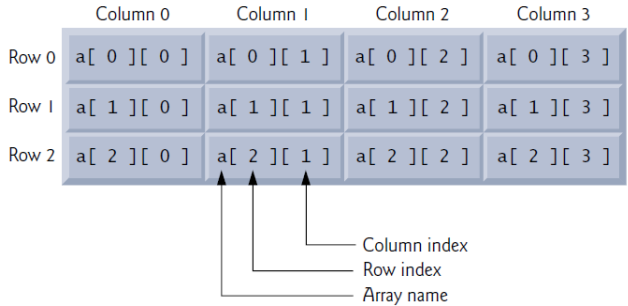
```c
#include <stdio.h>
#define SIZE 5

void modifyArray(int b[], size_t size);
void modifyElement(int e);

int main(void)
{
   int a[SIZE] = {0, 1, 2, 3, 4}; // initialize array a

   puts("Effects of passing entire array by reference:\n\nThe values of the original array are:");

   for (size_t i = 0; i < SIZE; ++i) {
      printf("%3d", a[i]);
   }

   puts(""); // outputs a newline
   modifyArray(a, SIZE); // pass array a to modifyArray by reference
   puts("The values of the modified array are:");

   // output modified array
   for (size_t i = 0; i < SIZE; ++i) {
      printf("%3d", a[i]);
   }

   printf("\n\n\nEffects of passing array element by value:\n\nThe value of a[3] is %d\n", a[3]);
   modifyElement(a[3]); // pass array element a[3] by value
   printf("The value of a[3] is %d\n", a[3]);
}

void modifyArray(int b[], size_t size)
{
   for (size_t j = 0; j < size; ++j) {
      b[j] *= 2; // actually modifies original array
   }
}
void modifyElement(int e)
{
   printf("Value in modifyElement is %d\n", e *= 2);
}
```

Tarea

- Consultar como funciona el algoritmo de ordenamiento de burbuja.
- Consultar como funciona la busqueda lineal vs la busqueda binaria en un arreglo.

Arreglos - Multidimensionales

Un arreglo multidimensional consiste en formar un vector de vectores.



La inicialización se realizar indicando las dos dimensiones entre dos corchetes para las filas y columnas. Asi

```
1  int b[2][2] = {{1}, {3, 4}};
2
```

Arreglos - Multidimensionales - Ejemplo

```c
#include <stdio.h>

void printArray(int a[][3]); // function prototype

// function main begins program execution
int main(void)
{
   int array1[2][3] = {{1, 2, 3}, {4, 5, 6}};
   puts("Values in array1 by row are:");
   printArray(array1);

   int array2[2][3] = {1, 2, 3, 4, 5};
   puts("Values in array2 by row are:");
   printArray(array2);

   int array3[2][3] = {{1, 2}, {4}};
   puts("Values in array3 by row are:");
   printArray(array3);
}

// function to output array with two rows and three columns
void printArray( int a[][3])
{
   // loop through rows
   for (size_t i = 0; i <= 1; ++i) {
      // output column values
      for (size_t j = 0; j <= 2; ++j) {
         printf("%d ", a[i][j]);
      }
      printf("\n"); // start new line of output
   }
}
```

El compilador necesita saber cuantos elementos hay en cada fila, por eso se requiere mandar el dato de las columnas cuando se hace el llamado a la función.

Arreglos - Multidimensionales - Ejemplo

Implemente un programa que dado una matriz de notas de estudiantes, obtenga la menor nota, la mayor nota, y el promedio de cada estudiante.

```
The array is:
                 [0]  [1]  [2]  [3]
studentGrades[0] 77   68   86   73
studentGrades[1] 96   87   89   78
studentGrades[2] 70   90   86   81

Lowest grade: 68
Highest grade: 96
The average grade for student 0 is 76.00
The average grade for student 1 is 87.50
The average grade for student 2 is 81.75
```

Arreglos - Longitud Variable

Son arreglos cuyo tamaño se elige en tiempo de ejecución y no en tiempo de compilación (Caracteristica no soportada en Microsoft Visual C++).

ARREGLOS EN C

GRACIAS