

LECCIÓN 0: Antecedentes

María de la Paz Guerrero Lebrero

Curso 2014 / 2015

Grado en Matemáticas

maria.guerrero@uca.es



Índice

- Introducción
- Vectores (1xn)
- Cadenas de caracteres.
- Funciones para cadenas de caracteres.
- Instrucciones de E/S de cadenas.

Introducción

- Datos estructurados
- Definición
- Representación

Datos estructurados

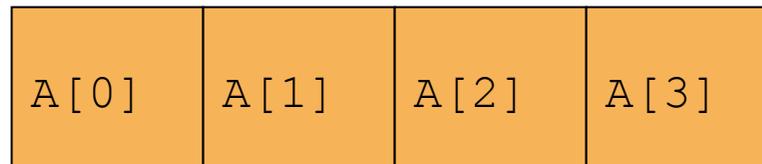
- Hasta ahora sólo hemos trabajado con datos simples que ocupan una posición de memoria:
 - Enteros (int)
 - Reales (float)
 - Caracteres (char)
 - Lógicos (bool)
- Para generar instrucciones complejas partiremos de los datos simples conocidos hasta ahora.

Definición

- Un **vector** es una estructura de datos formada por un conjunto de **datos del mismo tipo** agrupados bajo un **nombre común**.
- Declaración:
 - `<tipo> <nombre> [nº de eltos.]`
- Ejemplos:
 - `int A[10];`
 - `char dni[8];`
 - `float B[20];`

Representación

- Dada la declaración del vector A :
 - `int A[4];`



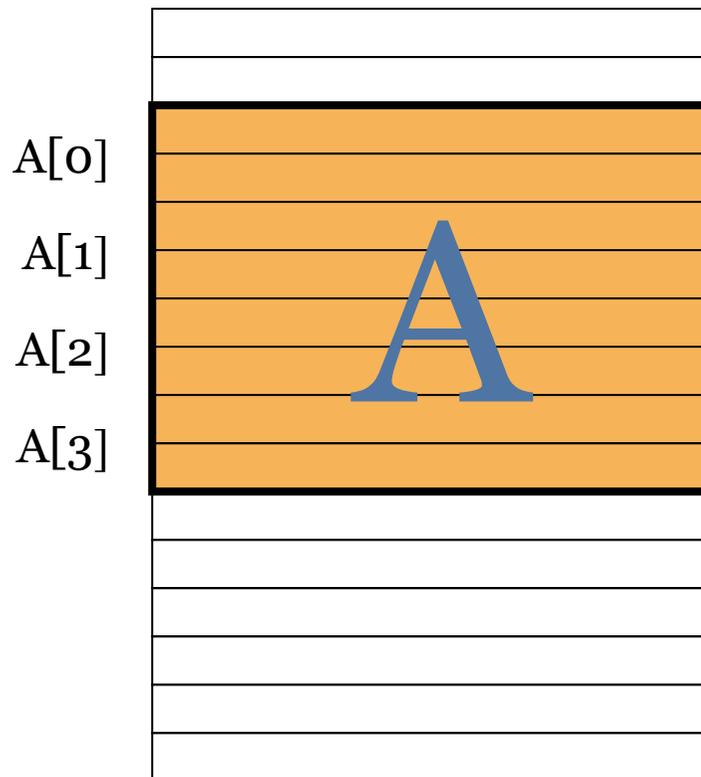
- El primer elemento del vector será $A[0]$ y el último elemento será $A[3]$.

Vectores

- Almacenamiento interno
- Acceso a un elemento
- Operaciones con vectores
- Recorridos de un vector
- Un ejemplo
- Ejercicios

Almacenamiento interno

```
int A[4];
```



- Los elementos de un vector ocupan posiciones de memoria consecutivas.
- Índices negativos o mayores que tres (en este caso) acceden a otras posiciones de memoria.

Acceso a un elemento

- El acceso a un elemento de un vector se realiza por su posición relativa al comienzo del mismo.
- Se accede a un elemento de un vector para:
 - leer su contenido ó
 - para asignarle un nuevo valor.

Acceso a un elemento

- Para acceder a un elemento de un vector se escribe el nombre se escribe el nombre de la variable y la posición que ocupa dicho elemento entre corchetes:
 - `<nombre variable> [<posición>]`
- Ejemplos:
 - `int a[3], b[30];`
 - `a[0] = 3.5;`
 - `for (i=0;i<30;i++) b[i]=0;`

Acceso a un elemento

- Error habitual: acceder a posiciones del vector incorrectas, debido a :
 - Indices menores que el menor índice permitido ($V[-3]$)
 - Indices mayores que el mayor índice permitido ($V[28]$)
- Un acceso “fuera de rango” produce efectos inesperados, tales como :
 - Cuelgue del ordenador
 - Bucles sin fin
 - Modificación de los valores de otras variables
 - Comportamientos extraños del programa

Operaciones con vectores

- Las operaciones con vectores permitidas dependen del lenguaje
- Normalmente, no se permite ninguna operación, y de existir alguna, ésta es la asignación de un vector a otro del mismo tipo
- En C no se permite ninguna operación sobre vectores completos

Recorridos de un vector

- Dado que el acceso a los elementos de un vector se hace elemento a elemento, y que se conoce a priori su tamaño, es muy frecuente acceder a sus elementos con un bucle FOR
- Ejemplo: Asignar 0 a todos los elementos de un vector de tamaño N

```
for (i=0;i<N;i++)  
    A[i]=0;
```

Recorridos de un vector

- Recorridos habituales de un vector:
 - Asignar a un vector de tamaño N los valores 1,2,...

```
for (i=0;i<N;i++)  
  A[i] = i+1;
```
 - Asignar valores aleatorios

```
for (i=0;i<N;i++)  
  A[i]=rand();
```
 - Rotar sus valores hacia la derecha

```
aux = A[N-1];  
for (i=N-1;i>0;i--) {  
  A[i]=A[i-1];  
}  
A[0]=aux;
```

Un ejemplo

- Media aritmética de un vector de enteros:

```
#include <stdio.h>

void main (void)
{
    int v[10] = {10, 2, 3, 5, 6, 8, 3, 5, 9, 7};
    int i;
    float media, suma = 0;

    for (i = 0; i <10; i++)
        suma = suma + v[i];

    media = suma / 10;
}
```

Ejercicios

1. Inicializa un vector con los 10 primeros números pares.
2. Inicializa un vector con los 10 primeros números impares.
3. Crea un nuevo vector que sea la suma de los dos vectores anteriores.
4. Busca un elemento que te dé el usuario en el vector del ejercicio 3.

Cadenas de caracteres

- Definición
- Funciones de cadenas de caracteres
- Funciones de E/S de cadenas de caracteres

Cadenas de caracteres

- No existe un tipo específico. Una cadena de caracteres es un vector unidimensional en el que sus elementos son caracteres.
- Forma genérica de su declaración:

```
char <nombre de la cadena> [<tamaño>];
```

- Se pueden inicializar como un vector normal o se les puede asignar una constante de tipo cadena de caracteres.

Cadenas de caracteres

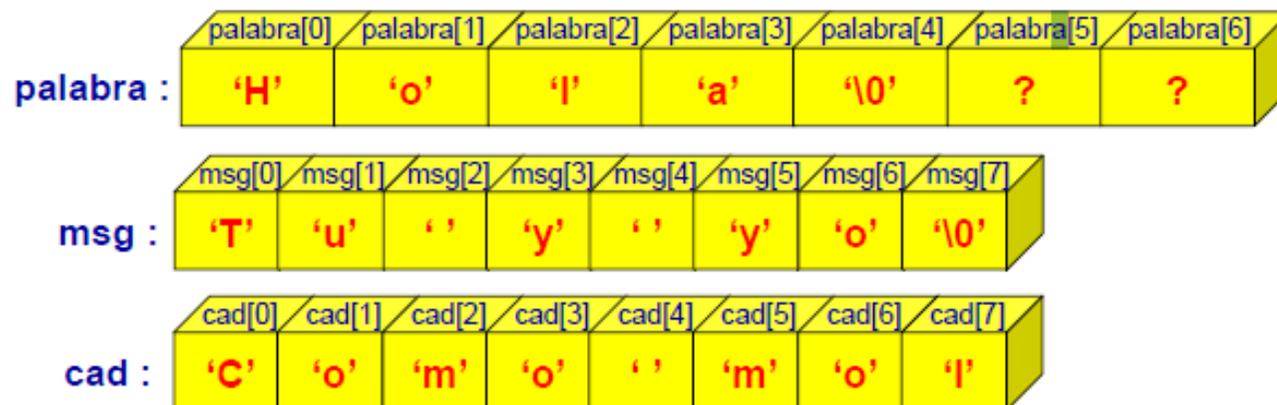
- Si no se especifica el tamaño de la cadena, este será el de la misma mas uno, ya que el compilador añade el carácter NULL(“\0”) al final de la cadena.

Cadenas de caracteres

- Ejemplos

```
char palabra [7] = "Hola";
char msg [ ] = "Tu y yo";
char cad [8] = "Como mola mi gramola";
```

La reserva de espacio para almacenar una cadena no impide que se pueda producir un desbordamiento del array.



Funciones de cadenas de caracteres

Función STRLEN

```
# include <string.h>  
int strlen (char* string);
```

- Devuelve la longitud de la cadena de caracteres que se le pasa como parámetro.
 - Si se ejecuta correctamente, devuelve la longitud de la cadena en bytes, sin incluir el ‘\0’.
 - Si se produce algún error, el valor devuelto es 0.

Funciones de cadenas de caracteres

- Ejemplo *strlen*

```
#include <stdio.h>
#include <string.h>

int main ()
{
    char cad[20] = "Hola mundo";
    int num;
    num = strlen(cad);
    printf("La longitud de la cadena es %d\n", num);
    return 0;
}
```

Funciones de cadenas de caracteres

Función STRCAT

```
# include <string.h>
```

```
char* strcat (char* str1, char* str2);
```

- Devuelve la concatenación de las cadenas de caracteres que se le pasa como parámetros.
 - Si se ejecuta correctamente, la función concatena str2 al final de str1.
 - Si se produce algún error, el valor devuelto es '\0'.

Funciones de cadenas de caracteres

- Ejemplo *strcat*

```
#include <stdio.h>
#include <string.h>

int main ()
{
    char cad1[20] = "Mortadelo";
    char cad2[20] = "Filemon";
    strcat(cad1, cad2);
    printf("La concatenacion es %s\n", cad1);
    return 0;
}
```

Funciones de cadenas de caracteres

Función STRCMP

```
# include <string.h>
```

```
char* strcmp (char* str1, char* str2);
```

- Devuelve la comparación de las cadenas de caracteres que se le pasa como parámetros.
 - **Si se ejecuta correctamente:**
 - Devuelve un valor menor que 0, si str1 es menor que str2.
 - Devuelve el valor 0 si las cadenas son iguales.
 - Devuelve un valor mayor que 0, si str1 es mayor que str2.

Funciones de cadenas de caracteres

- Ejemplo *strcmp*

```
#include <stdio.h>
#include <string.h>

int main ()
{
    char cad1[20] = "Hola";
    char cad2[20] = "Adios";
    if(strcmp(cad1, cad2) == 0)
        printf("Son iguales\n");
    else
        printf("Son distintas\n");
    return 0;
}
```

Funciones de cadenas de caracteres

Función STRCPY

```
# include <string.h>  
char* strcpy (char* str1, char* str2);
```

- Devuelve una copia de la cadena que se le pasa como parámetro.
 - Si se ejecuta correctamente, copia el contenido de str2 en str1 incluido el '\0'.
 - Si se produce algún error, devuelve NULL.

Funciones de cadenas de caracteres

- Ejemplo *strcpy*

```
#include <stdio.h>
#include <string.h>

int main ()
{
    char cad1[20] = "Hola";
    char cad2[20];
    strcpy(cad2, cad1);
    printf("La copia es %s\n", cad2);
    return 0;
}
```

Funciones de E/S de cadenas de caracteres

Función GETS

```
# include <stdio.h>  
  
char* gets (char* string);
```

- Lee una línea de la entrada estándar y la almacena en la cadena que se le pasa como parámetro.
 - Si se ejecuta correctamente, almacena la línea en la cadena sustituyendo el carácter ‘\n’ por el ‘\0’.
 - Si se produce algún error, devuelve NULL.

Funciones de cadenas de caracteres

- Ejemplo *gets*

```
#include <stdio.h>

int main ()
{
    char cadena[20];
    printf("Introduzca una cadena de texto:");
    gets(cadena);
    printf("La cadena es %s\n", cadena);
    return 0;
}
```

Funciones de E/S de cadenas de caracteres

Función PUTS

```
# include <stdio.h>  
char* puts (char* string);
```

- Escribe la cadena que se le pasa como parámetro en la salida estándar.
 - Si se ejecuta correctamente, sustituye el carácter ‘\o’ por el ‘\n’ y devuelve un valor positivo.
 - Si se produce algún error, devuelve EOF.

Funciones de cadenas de caracteres

- Ejemplo *puts*

```
#include <stdio.h>

int main ()
{
    char cadena[20];
    printf("Introduzca una cadena de texto:");
    gets(cadena);
    puts("Cadena adquirida");
    return 0;
}
```

Funciones de E/S de cadenas de caracteres

Función GETCHAR

```
# include <stdio.h>  
  
int getchar ();
```

- Lee un carácter de la entrada estándar.
 - Si se ejecuta correctamente, devuelve el entero correspondiente al carácter leído.
 - Si se produce algún error, devuelve EOF.

Funciones de cadenas de caracteres

- Ejemplo *getchar*

```
#include <stdio.h>

int main ()
{
    char cad[20], c;
    int i = 0;
    while((c=getchar()) != '$'){
        cad[i]= c;
        i++;
    }
    return 0;
}
```

Funciones de E/S de cadenas de caracteres

Función PUTCHAR

```
# include <stdio.h>  
  
int putchar (int character);
```

- Escribe un carácter en la salida estándar.
 - Si se ejecuta correctamente, devuelve el carácter escrito en la entrada estándar.
 - Si se produce algún error, devuelve EOF.

Funciones de cadenas de caracteres

- Ejemplo *putchar*

```
#include <stdio.h>
#include <string.h>
int main ()
{
    char cad[20] = “La asignatura de Informatica es
                    genial ”;

    int i;
    for(i =0; i< strlen(cad);i++){
        putchar(cad[i]);
    }
    return 0;
}
```

Ejercicios

1. Dada una cadena de caracteres, añadir eco a la misma, conociendo el número de veces que se repita la última letra.
2. Dada una cadena de caracteres, eliminar los espacios en blanco desplazando los caracteres a la izquierda.
3. Dada una cadena de caracteres, contar el números de vocales, de consonantes y de dígitos.