UNIVERSIDAD DE LOS ANDES
MERIDA VENEZUELA

Unidad 3. Estructuras simples de datos

Tema 1. Cadenas de caracteres

UNIVERSIDAD DE LOS ANDES

Cadenas de Caracteres

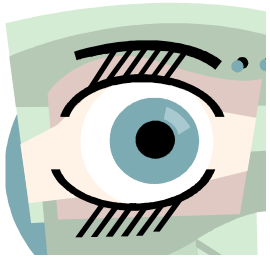
- Contenido:
  - Conceptos básicos
  - Representación algorítmica
  - Codificación
- Objetivo:

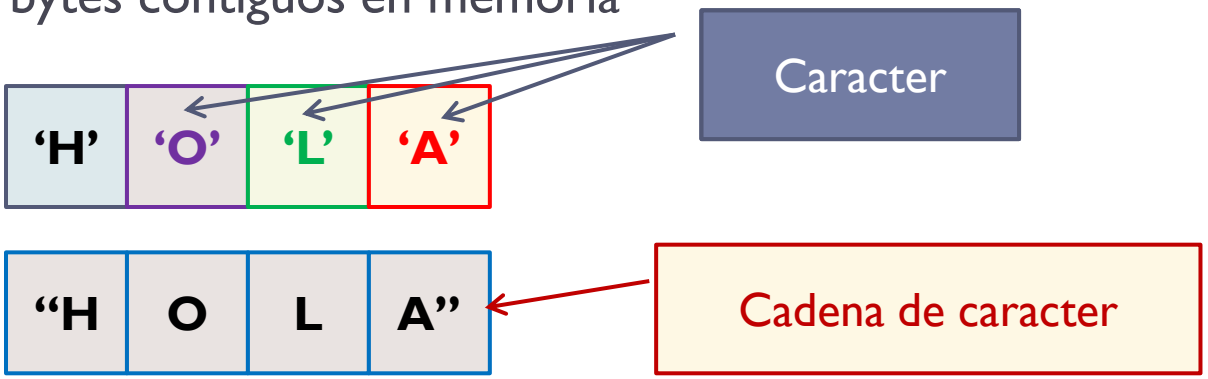
  Desarrollar habilidades en el uso de las cadenas de caracteres
- Bibliografía:
  - Deitel y Deitel, cap. 8.
  - http://www.ing.ula.ve/~ibc/pr1
  - http://www.webdelprofesor.ula.ve/ingenieria/ibc
  - Joyanes, cap. 7.

UNIVERSIDAD DE LOS ANDES

Cadenas de caracteres

- es una secuencia de caracteres definida en un alfabeto
- Alfabeto: conjunto de caracteres disponibles
- Ejemplo: Alfabeto = ASCII, cadenas de caracteres en ese alfabeto = "hola", "casa", "#$ok)*&"
- Estructura de almacenamiento
  - vector de caracteres
  - Un byte (8 bits) por cada caracter de la cadena si el alfabeto es ASCII, almacenados en bytes contiguos en memoria





UNIVERSIDAD DE LOS ANDES

Operaciones

- **longitud(cadena): Entero+,** regresa la longitud actual de la cadena

  longitud("casa") regresa 4

- **concatenar(cadena1, cadena2): Cadena,** concatena dos cadenas, coloca cadena1 seguida de cadena2

  concatenar("casa","blanca") regresa "casablanca"

- **subcadena(cadena, inicio, longitud):cadena,** regresa la subcadena que comienza en la posición inicio y que tiene la longitud solicitada

  subcadena("casablanca", 3, 2) regresa "sa"

  subcadena("miraflores", 4, 5) regresa "aflor"

  subcadena("Reunion hoy a las 4:15 pm", 19, 4) regresa "4:15"

UNIVERSIDAD DE LOS ANDES

Operaciones

- **indice(cadena1, cadena2): Entero+,** regresa la posición de inicio de la primera ocurrencia de la cadena2 dentro de la cadena1

  indice("casablanca","a") regresa 2

- **las comparaciones (< , >, =, ≤ , ≥ , ≠) se efectúan según el orden lexicográfico dado por el alfabeto que se esté utilizando como el código ASCII o UNICODE**

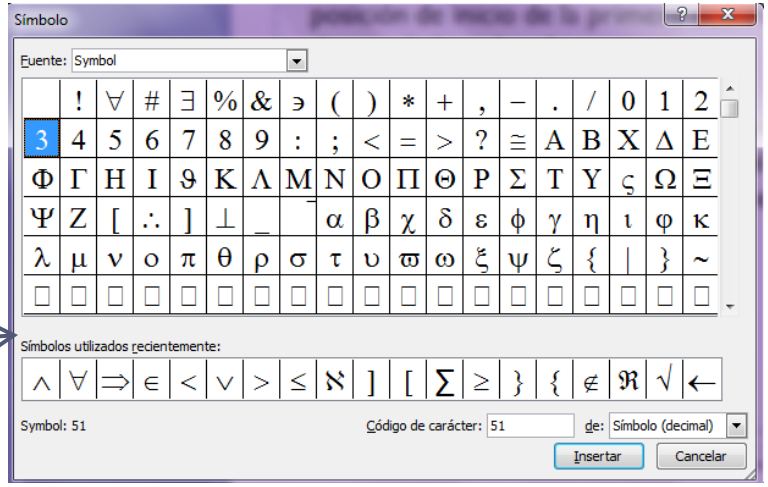
  "casa" < "papa" **cierto**

  "minas" <= "hechos" **cierto**

  "cascara" = "cascaras" **falso**

  "viento" ≠ "vientos" **cierto**

"3(6)+" > "3[6]+" falso



UNIVERSIDAD DE LOS ANDES

TAD Cadena

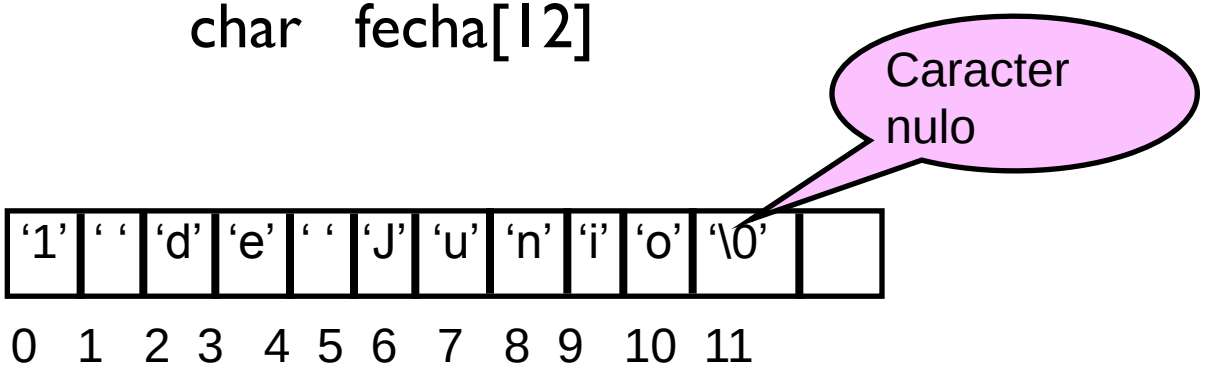
05/07/11

Especificación (7) Cadena[DIM]De Caracter

1	Sintáctica: creaCadena(Entero+) -> Cadena[DIM], longitud(Cadena[DIM]) -> Entero+, concatena(Cadena[DIM], Cadena1[DIM1]) -> Cadena[DIM+DIM1], subcadena(Cadena[DIM], ini, long) -> Cadena1[longi], indice(Cadena[DIM], Cadena1[DIM1]) -> Entero+, Cadena[DIM] < Cadena1[DIM1] -> Lógico, Cadena[DIM] > Cadena1[DIM1] -> Lógico, Cadena[DIM] = Cadena1[DIM1] -> Lógico, Cadena[DIM] ≤ Cadena1[DIM1] -> Lógico, Cadena[DIM] ≥ Cadena1[DIM1] -> Lógico, Cadena[DIM] ≠ Cadena1[DIM1] -> Lógico.	creaCadena(): Crea un vector de caracteres vacío, sin limpiar la memoria. longitud(): Devuelve el número de elementos actuales en el vector. concatena(): Devuelve la cadena de caracteres que contiene la cadena seguida de cadena1. subcadena(): Regresa la subcadena que comienza en ini con long caracteres, si long < DIM, sino regresa el final de la cadena que comienza en ini. indice(): Regresa la posición de inicio de la primera ocurrencia de cadena1 dentro de la cadena, si existe, sino regresa 0. **<** , **>** , **=** , **≠** , **≤** , **≥** : Comparadores de cadenas según el orden lexicográfico de los caracteres en el alfabeto
2	Declaraciones: i, l, dim, dim1: Entero+ c: Cadena[DIM]	
3	Semántica: longitud(creaCadena(dim)) = 0 longitud(concatena(creaCadena(dim), creaCadena(dim1))) = 0 indice(creaCadena(dim), creaCadena(dim1)) = 0 concatena(c, creaCadena(dim)) = c creaCadena(dim) < creaCadena(dim1) = Falso creaCadena(dim) > creaCadena(dim1) = Falso creaCadena(dim) = creaCadena(dim1) = Verdadero	

Cadenas de caracteres en C/C++

- Una cadena de caracteres (*string*) es un conjunto de caracteres (incluido el blanco) que se almacenan en localidades contiguas de memoria.
- Se representa como un vector de caracteres donde cada elemento del vector representa un caracter de la cadena.
- Ejemplo char fecha[12]



Nótese que una cadena de n caracteres requerirá un vector de n+1 elementos, debido al caracter nulo '\0'que se añade automáticamente al final de la cadena

Declaración e inicio de una cadena de caracteres en C/C++

- Notación algorítmica

Cadena nombre[dim]

- C/C++

char nombre[dim];

- donde dim es el número de caracteres de la cadena + 1
- Ejemplos:

  Cadena linea[81]

  char linea[81];

  Cadena color[10]

  char color[10];

#define MAX 256

char palabra[MAX];

- declara una cadena llamada palabra con un máximo de 255 caracteres

char s[MAX]={'H','o','l','a','\0'};

- declara s con un máximo de 255 caracteres, pero se inicia con 4 caracteres

char sal[]="Hola";

- declara sal con el máximo de caracteres que tenga la cadena de inicio Hola, 4+1=5 caracteres

- Se accede por subíndice

sal[0]='H', sal[1]='o', sal[4]='\0'

Definición del tipo de dato Cadena en C/C++

#define MAX 256

typedef char Cadena[MAX];

Cadena palabra;

- para declarar un nuevo tipo de dato en C/C++ se utiliza **typedef**

typedef int edad;

edad hembra, varon;

typedef float altura;

altura hombre, mujer;

cin>>palabra; **Leer una**

gets(palabra); **cadena**

⇦se define el máximo MAX

⇦define el tipo Cadena

⇦declara la variable palabra

Ejemplos:

#define NDIAS 7

#define CAR 10

typedef char Cadena[CAR];

Cadena diasSemana[NDIAS]={"lunes", "martes", "miercoles", "jueves", "viernes", "sabado", "domingo"};

- **Escribir una cadena**

cout<<palabra<<endl;

puts(palabra);

EISULA. Departamento de Computación. Programación I. Sec 2. Isabel Besembel 22/2/2012

UNIVERSIDAD DE LOS ANDES

Funciones de librería para cadenas

Librería	Directiva	Descripción
string	#include < string.h>	Contiene los prototipos de las funciones y macros de clasificación de cadenas de caracteres

Longitud de una cadena (**strlen**): Devuelve la longitud de una cadena, equivalente a longitud(cadena): Entero

```
lon = strlen(palabra);
```

Asignación (**strcpy**): Copia la segunda cadena en la primera cadena, equivalente a cadena1=cadena2

```
strcpy (cadena1, cadena2); // cadena1 = cadena2
```

*Comparación (**strcmp**):* Compara dos cadenas. Si son iguales devuelve 0; si la primera es menor que la segunda devuelve un valor < 0; si la primera es mayor que la segunda devuelve un valor > 0. Equivalente a cadena1{=|<|>}cadena1

```
if (strcmp(cadena1, cadena2) == 0)          // Son iguales
if (strcmp(cadena1, cadena2) < 0)   // cadena1 < cadena2
if (strcmp(cadena1, cadena2) > 0)   // cadena1 > cadena2
```

UNIVERSIDAD DE LOS ANDES

Funciones de string.h

- **strcat(cad1, cad2)** regresa una cadena que contiene cad1 concatenada con cad2, equivale a concatena(cad1, cad2)

```
printf( "Ingrese su nombre: " );
scanf( "%s", nombre );
titulo = strcat( nombre," El Grande" );
printf( "Hola, %s\n", titulo );
```

- **strncat(cad1, cad2, cont)** regresa una cadena de caracteres con cad1 concatenada con los primeros cont caracteres de cad2
- **strcspn(cad,1 cad2)** regresa la posicion de la primera ocurrencia de cualquier caracter de cad1 en cad2

```
pos = strcspn("que tal","a")        regresa 6
```

- **strncpy(cad1, cad2, cont)** regresa la cadena con cad1 con al menos cont caracteres copiados de cad2 a cad1, si cad2 tiene menos de cont caracteres, coloca al final '\0'

UNIVERSIDAD DE LOS ANDES

Ejercicio resuelto 1

- Enunciado del problema

  Encontrar todas las ocurrencias de una letra, por ejemplo 'p', en un texto dado

- Análisis E-P-S

Entrada	Variable	Descripción	Tipo de dato	Rango válido
	letra texto	Letra que se buscara en el texto Texto que contiene un valor predeterminado de caracteres	Caracter Cadena de caracteres	Alfabeto ASCII [0,MAX] caracteres en el alfabeto ASCII
Proceso	Obtener el texto y la letra a buscar Repetir hasta que texto esté vacío [Buscar letra en texto Si se encuentra, guardar su posición en el vector posi Cortar texto quitando hasta la ultima letra encontrada]			
Salida	Mensaje indicando posi	Vector de las posiciones de la letra en texto	Vector[MAX] De Entero	pos(i) $\in$ [0, MAX]

Ejercicio resuelto 1. Diseño

	todasLasOcurrencias {pre: }	{pos: texto = “” }
1 2 3 4 5 6	Desplegar (“Introduzca el texto y la letra”) texto, letra = valor suministrado Desplegar (texto) i=1 (lon(texto)≠0)[pos=indice(texto, letra) Si(pos≠0) entonces posi[i], i=pos, i+1 [texto[k]=texto[pos+k]] k=1,lon(texto)-pos sino texto=“” fsi] Si(i=0) entonces Desplegar (“ no tiene ninguna ”, letra) Sino Si(i=1) entonces Desplegar(“ tiene una sola ”, letra, “en”, posi[i]) Sino Desplegar(“ tiene ”, letra, “ en ”) [Desplegar posi[k]] k=1, i Fsi	● **letra:** Caracter. Valor a buscar. ● **texto:** Cadena. Valor dentro del que se buscan todas las ocurrencias de letra. ● **pos:** Entero. Posición de la primera ocurrencia de letra dentro de texto. ● **indice()**. Función que calcula la posición de la primera ocurrencia de un caracter dentro de una cadena de caracteres. ● **posi:** VectorDe Entero. Vector que contiene la posición de las ocurrencias de letra dentro de texto, si las hay. ● **i, k:** Entero. Subíndices
1 2 3	texto=“scdgtopky”, letra=‘t’ => posi=[5] texto = “as”, n = ‘p’ => posi = [] texto=“ppp”, letra=‘p’ => posi=[1, 2, 3]	● Caso con 1 ocurrencia ● Caso sin ocurrencia ● Caso con todas las ocurrencias posibles

Ejercicio resuelto 1. Implementación

```
/*          21/2/12                 Ejercicio resuelto 1
            Clase: 11
            Encontrar todas las ocurrencias de una letra dada, por ejemplo 'p',
            en un texto leido desde el teclado
*/
#include<stdio.h>
#include<string.h>

int main()
{
            char        letra[2];
            int         n, pos, i=0, k, lonTexto;

            printf("Ingrese el valor de la longitud del texto\n");
            scanf("%i", &n);
            char        texto[n];
            int         posi[n];
            printf("Ingrese el texto de %i caracteres sin espacios intermedios\n", n);
            scanf("%s", &texto);
            printf("Ingrese la letra que desea encontrar en el texto\n");
            scanf("%s", &letra);
            printf("\n\nEl texto <%s> \n", texto);
            lonTexto=strlen(texto);
```

Ejercicio resuelto 1. Implementación

```
do
{
        pos=strcspn(texto, letra);
        if(pos<lonTexto)
        {
                posi[i]=pos;
                i=i+1;
                for(k=0;k<lonTexto-pos;k++)        texto[k]=texto[pos+k+1];
        }
        else
                texto[0]='\0';
        lonTexto=strlen(texto);
}while(lonTexto!= 0);
if(i==0)
        printf("no contiene ninguna <%s> ... ", letra);
else if(i==1)
        printf("contiene una sola <%s> en la posicion %i \n", letra, posi[0]);
else
{       printf("contiene <%s> en las posiciones \n", letra);
        for(k=0;k<i;k++)        printf(" %i ", posi[k]);
}
printf("\n.... Terminado .... \n");
return 0;
}
```

Encuentre otra solución equivalente

UNIVERSIDAD DE LOS ANDES

Pase de vectores como parámetros en C/C++

- Un vector completo se puede pasar a una función como parámetro
- Para pasar un vector como parámetro actual a una función, se especifica únicamente su nombre, sin corchetes ni subíndices.

```
char  caracteres[80] = "esta cadena es constante";
……
F1(caracteres, longitud);
……
```

- El parámetro formal debe ser definido dentro de la función, se escribirá un par de corchetes vacíos, es decir, el tamaño del vector no se especifica.

```
void F1  (char string[], int lng)
```

UNIVERSIDAD DE LOS ANDES

Pase de vectores como parámetros en C/C++

- En C/C++ los arreglos en general son pasados como parámetros por referencia. Esto es, el nombre del arreglo es la dirección del primer elemento del arreglo
- En C/C++ un elemento cualquiera de un arreglo puede ser pasado a una función por valor o por referencia, tal y como se hace con una variable simple

```
float     media(int a, float x[])     // Definición de la función
{                                     // Note que se incluyen los corchetes vacíos
    …….
}
void    main ()
{   int n;
    float med, lista[100];
    ……
    med = media(n, lista);            // Esta llamada pasa como parámetros actuales la longitud
                                      // del vector y el vector
    ……..                              // Note que no se incluyen los corchetes
}
```

- ¿Qué es una cadena de caracteres?
- ¿Cuál es la diferencia entre una cadena de caracteres y un caracter?
- ¿ Cuáles son las operaciones de las cadenas de caracteres?
- ¿Cómo se implantan las cadenas de caracteres en C/C++?
- ¿Cuáles son las operaciones del TAD Cadena?
- ¿Cuáles son las funciones de librería de C/C++ para las cadenas de caracteres?

Resumen

¿Cuáles son los conceptos relevantes de esta clase?

UNIVERSIDAD DE LOS ANDES

Ejercicios

- **Para cada uno de los enunciados dados a continuación realizar: Análisis en E-P-S, diseño en pseudocódigo y codificación en C o C++ utilizando funciones de librería**

1. Escribir un programa en C++ que lea una línea de texto, la almacene en un vector y la escriba al revés. La longitud de la línea no será especificada (terminará al pulsar la tecla Enter), pero se supone que no excederá de 80 caracteres.
2. Escribir la declaración de las siguientes variables:

   - Una variable cadena cad con un máximo de 20 caracteres

   - La misma variable cad iniciada con el valor "Caracas"

   - Dos variables nombre y apellido de tipo Cadena

   - Un vector nombres de tipo Cadena

   - Un vector meses iniciado con los nombres de los doce meses del año.

   - Un vector dias_semana iniciado con los nombres de los siete días de la semana

UNIVERSIDAD DE LOS ANDES

Ejercicios

4. Escribir una función

   logico busca_caracter(Cadena cad, char c)

   que busque un carácter c en una cadena cad, devolviendo un valor lógico Cierto si lo encuentra y Falso en caso contrario.
5. Escribir una función

   int busca_caracter(Cadena cad, char c)

   que cuente el número de apariciones de un carácter c en una cadena cad, devolviéndolo como resultado.
6. Escribir una función que reciba como parámetro una cadena de caracteres y elimine los espacios en blanco del final de la cadena
7. Dada una cadena de caracteres de cualquier longitud, cópiela en un arreglo de 20 caracteres de ancho (20 columnas) sin cortar ninguna palabra y utilizando las filas que sean necesarias.

   Ejemplo: "Este es un texto de ejemplo"

   Debe quedar como: "Este es un texto de "

   "ejemplo "

UNIVERSIDAD DE LOS ANDES

Ejercicios

8. Escribir un procedimiento

   void dia_semana(int dia, Cadena nomdia)

   que reciba como parámetro de entrada un número del 1 al 7 que representa el día de la semana (1=lunes, 2=martes, 3=miercoles, 4=jueves, 5=viernes, 6=sabado, 7=domingo) y devuelva en la cadena nomdia el nombre de dicho día. Utilice una tabla dias_semana.

9. Escribir un procedimiento que lea una palabra de hasta 20 caracteres y la escriba como se ve en la figura:

   Entrada: HOLA

   Salida:
   ```
   HOLA
   O  L
   L  O
   ALOH
   ```