

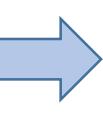


Facultad de Ingeniería Industrial y de Sistemas

Algoritmia y Estructura de Datos: Cadena de caracteres

Variable tipo char

El conjunto de
caracteres



Los caracteres son el conjunto de símbolos necesario para interactuar con un Sistema de cómputo.

Un conjunto de caracteres contiene letras mayúsculas y minúsculas, dígitos, signos de puntuación, operadores matemáticos y otros símbolos

Diferentes
conjuntos de
caracteres



La cantidad de símbolos de un conjunto de caracteres depende del número de bits que se utilizan para representarlos

Sistema Baudot	5 bits	2^5 caracteres
Sistema ASCII	7 bits	2^7 caracteres
Sistema ASCII extendido	8 bits	2^8 caracteres
Sistema EBCDIC	8 bits	2^8 caracteres
Sistema UNICODE, hasta 32 bits	2^{32}	caracteres

Variable tipo char

Diferentes conjuntos de caracteres



La cantidad de símbolos de un conjunto de caracteres depende del número de bits que se utilizan para representarlos

Sistema Baudot	5 bits	2^5 caracteres
Sistema ASCII	7 bits	2^7 caracteres
Sistema ASCII extendido	8 bits	2^8 caracteres
Sistema EBCDIC	8 bits	2^8 caracteres
Sistema UNICODE, hasta 32 bits	2^{32}	caracteres

Importante en cualquier conjunto de caracteres



Todo carácter tiene una posición en el conjunto
Las mayúsculas son caracteres consecutivos ('A', 'B', ..., 'Z')
Las minúsculas son caracteres consecutivos ('a', 'b', ..., 'z')
Los dígitos son caracteres consecutivos ('0', '1', ..., '9')
Un carácter es mayor que otro porque ocupa una posición mayor

Variables carácter y cadena de caracteres

‘x’
‘9’
‘+’ } Son caracteres

“UNI”
“Covid 19”
“2x+3y” } Son cadenas de caracteres

Declaración en C++

```
char simbolo;
```

asignación

```
simbolo = ‘a’ ;  
texto[18] = ‘b’;  
texto[7] = ‘9’;  
texto[i] = simbolo;  
simbolo = texto[12];
```

Declaración en C++

```
char texto[20];
```

Número de elementos de la cadena texto

lectura

```
cin>>simbolo;  
cin>>texto;
```

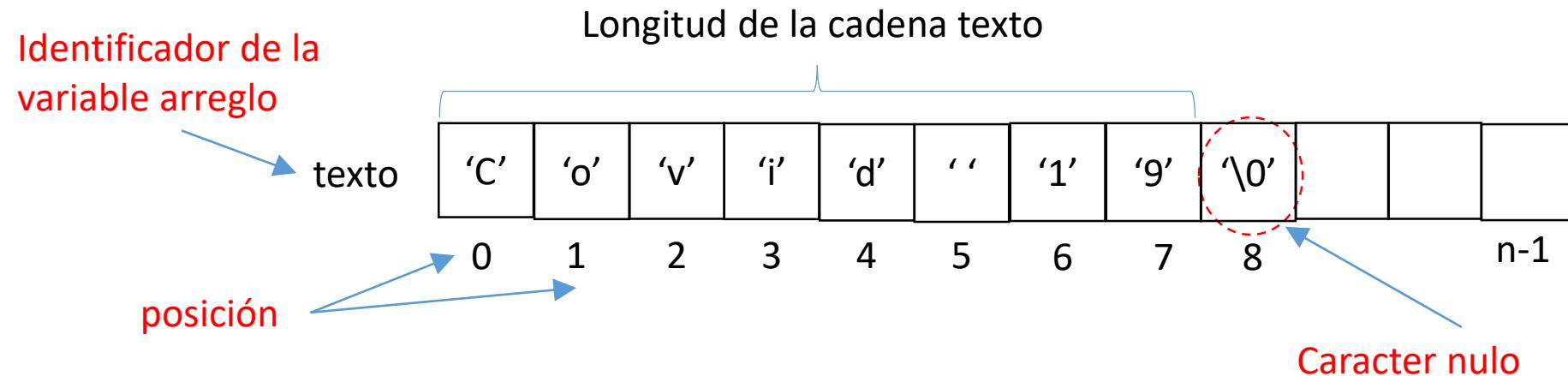
escritura

```
cout<<simbolo;  
cout<<texto;  
cout<<texto[9];
```

Se lee toda la cadena texto

Se escribe toda la cadena texto

Cadena de caracteres



El carácter nulo marca el final de la cadena.

La lectura de una cadena coloca los caracteres en el arreglo de caracteres y agrega el carácter nulo

La escritura escribe todos los caracteres anteriores al carácter nulo

Problema básico: la longitud de la cadena

Desarrollar un algoritmo que lea una palabra y determine cuántos caracteres tiene.

```
#include<iostream>
using namespace std;
main() {
    char palabra[30];
    int i;
    cout<<"Ingrese una palabra: ";
    cin>>palabra;
    i=0;
    while(palabra[i]!='\0')
        i++;
    cout<<"Tiene "<<i<<" caracteres"<<endl;
}
```

Problema básico: la longitud de la cadena

Desarrollar un algoritmo que lea una oración y determine cuántos caracteres tiene.

Diferencia entre **cin** y **gets** al leer una cadena

```
#include<iostream>
using namespace std;
main() {
    char oración[100];
    int i;
    cout<<"Ingrese una oracion: ";
cin>>oracion;
    i=0;
    while(oracion[i]!='\0')
        i++;
    cout<<"Tiene "<<i<<" caracteres"=<<endl;
}
```

```
#include<iostream>
using namespace std;
main() {
    char oracion[100];
    int i;
    cout<<"Ingrese una oracion: ";
gets(oracion);
    i=0;
    while(oracion[i]!='\0')
        i++;
    cout<<"Tiene "<<i<<" caracteres"=<<endl;
}
```

Ejemplo, si oración es:
UNI solo para emprendedores
Da como resultado:
Tiene 3 caracteres

Ejemplo, si oración es:
UNI solo para emprendedores
Da como resultado:
Tiene 27 caracteres

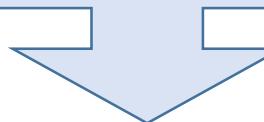
Problema básico: la longitud de la cadena

Desarrollar un algoritmo que lea una oración y determine cuántos caracteres tiene.

Uso de **cin.getline** para leer una cadena

```
#include<iostream>
using namespace std;
main() {
    char oracion[100];
    int i;
    cout<<"Ingrese una palabra: ";
    cin.getline(oracion,20);
    i=0;
    while(oracion[i]!='\0')
        i++;
    cout<<"Tiene "<<i<<" caracteres"=<<endl;
}
```

```
#include<iostream>
using namespace std;
main() {
    char oracion[100];
    int i;
    cout<<"Ingrese una palabra: ";
    cin.getline(oracion,100);
    i=0;
    while(oracion[i]!='\0')
        i++;
    cout<<"Tiene "<<i<<" caracteres"=<<endl;
}
```

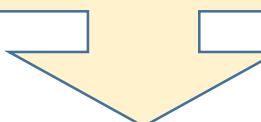


Ejemplo, si oración es:

UNI solo para emprendedores

Da como resultado:

Tiene 19 caracteres



Ejemplo, si oración es:

UNI solo para emprendedores

Da como resultado:

Tiene 27 caracteres

Problema sobre cadenas (conteo de vocales)

Tarea

Desarrollar un algoritmo que lea una oración y determine la cantidad total de vocales que tiene. Asuma que no hay tildes.

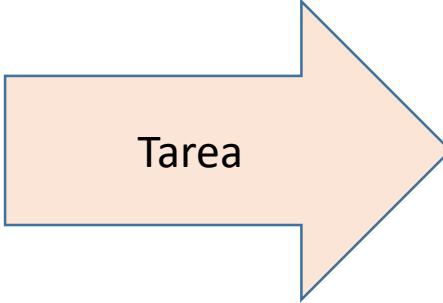
Ejemplo:
UNI solo para emprendedores

Total de vocales de la oracion es 11

Código

```
1 #include <iostream>
2 using namespace std;
3 main(){
4     char oracion[50];
5     int cont=0,i=0;
6     cout<<"Ingrese una oracion: ";
7     gets(oracion);
8     while (oracion[i]!='\0'){
9         switch(oracion[i]){
10             case 'a':
11             case 'A':
12             case 'e':
13             case 'E':
14             case 'i':
15             case 'I':
16             case 'o':
17             case 'O':
18             case 'u':
19             case 'U': cont++;
20         }
21         i++;
22     }
23     cout<<"La oracion "<<cont<<" vocales"<<endl;
24 }
```

Problema sobre cadenas (conteo de palabras)



Tarea

Desarrollar un algoritmo que lea una oración y determine cuántas palabras tiene.

Estrategia 1: detectar el inicio de cada palabra

Una palabra empieza cuando el carácter es diferente de blanco y el anterior es blanco. Si el primer carácter es diferente de blanco también empieza una palabra.

Estrategia 2: detectar el final de cada palabra.

Una palabra termina si el carácter es diferente de blanco y el siguiente es blanco o nulo.

Ejemplo:
UNI solo para emprendedores

Tiene 4 palabras

Problema sobre cadenas (conteo de palabras, solución 1)

Tarea

Desarrollar un algoritmo que lea una oración y determine cuántas palabras tiene.

Ejemplo:

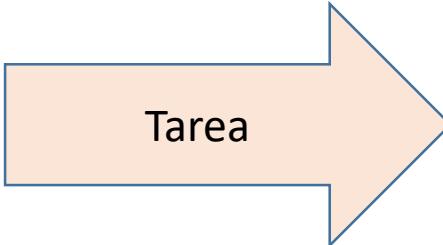
UNI solo para emprendedores

Tiene 4 palabras

Solución detectando el inicio de cada palabra (estrategia 1)

```
1 #include <iostream>
2 using namespace std;
3 main(){
4     char oracion[100];
5     int conpal=0,i=0;
6     cout<<"Ingrese una oracion: ";
7     gets(oracion);
8     if (oracion[0]!=' ')
9         conpal=1;
10    i=1;
11    while (oracion[i]!='\0'){
12        if (oracion[i]!=' ' && oracion[i-1]==' ')
13            conpal++;
14        i++;
15    }
16    cout<<"La oracion tiene "<<conpal<<" palabras" << endl;
17 }
```

Problema sobre cadenas (conteo de palabras, solución 2)

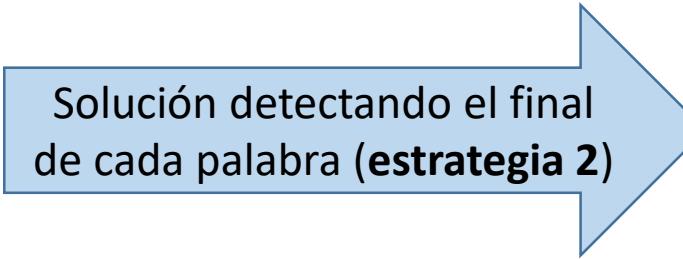


Tarea

Desarrollar un algoritmo que lea una oración y determine cuántas palabras tiene.

Ejemplo:
UNI solo para emprendedores

Tiene 4 palabras



Solución detectando el final
de cada palabra (**estrategia 2**)

```
1 #include <iostream>
2 using namespace std;
3 main(){
4     char oracion[100];
5     int conpal=0,i=0;
6     cout<<"Ingrese una oracion: ";
7     gets(oracion);
8     while (oracion[i]!='\0'){
9         if ((oracion[i+1]==' ' || oracion[i+1]=='\0') && oracion[i]!=' ')
10            conpal++;
11         i++;
12     }
13     cout<<"La oracion tiene "<<conpal<<" palabras" << endl;
14 }
```

Posición de un carácter en el sistema de caracteres ASCII

variable tipo int = variable tipo char

variable tipo char = variable tipo int

Cuando una variable int recibe el valor de una variable char, guarda la posición del carácter en el Sistema de caracteres

Cuando una variable char recibe un valor tipo int, guarda el carácter que corresponde al valor de la variable tipo int en el Sistema de caracteres

```
#include <iostream>
using namespace std;
main() {
    char car;
    int pos;
    car = 'A';
    pos = car;
    cout<<car<<" tiene la posicion "<<pos<<endl;
    pos=66;
    car = pos;
    cout<<"La posicion 66 corresponde al carácter "<<car<<endl;
}
```

Problema sobre conversión de carácter a número

Tarea

Dado un número real determinar cuánto suman sus cifras

Ejemplo:

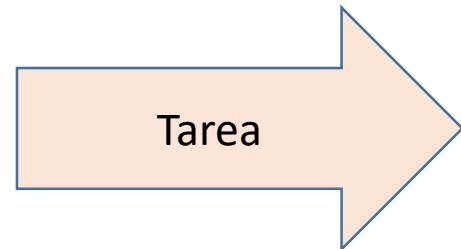
Número: **24567.359**

La suma de los dígitos es 41

Estrategia

Leer el número real como cadena y luego detectar los dígitos, convertirlos a número y sumarlos.

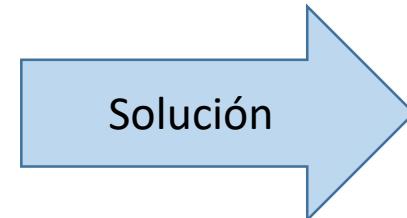
Problema sobre conversión de carácter a numero



Dado un número real determinar cuánto suman sus cifras

Ejemplo:
fecha: 24567.359

La suma de los dígitos es 41



```
1 #include <iostream>
2 using namespace std;
3 main(){
4     char numreal[50];
5     int i,suma=0, digito, digcero='0';
6     cout<<"Ingrese un numero real: ";
7     gets(numreal);
8     i=0;
9     while (numreal[i]!='\0'){
10        if (numreal[i]>='0' && numreal[i]<='9'){
11            digito=numreal[i];
12            suma=suma+digito-digcero;
13        }
14        i++;
15    }
16    | cout<<"La suma de los digitos es "<<suma<<endl;
17 }
```

Problema sobre cadenas (conteo de palabras que cumplen una condición)

Tarea

Desarrollar un algoritmo que lea una oración y determine cuántas palabras tienen una longitud dada.

Ejemplo:

Oración: **UNI solo para emprendedores**

Longitud dada: 4

Hay 2 palabras de longitud 4

Estrategia

Detectar el inicio y final de cada palabra y determinar la longitud por diferencia de las posiciones.

oración

'U'	'N'	'I'	' '	's'	'o'	'I'	'o'	' '	'p'	'a'	'r'	'a'	' '	'g'	'a'	'n'	'a'	'd'	'o'	'r'	'e'	's'	'\n'	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

↓
inicio

↓
final

$$\text{Longitud de palabra} = \text{final} - \text{inicio} + 1$$

Problema sobre cadenas (conteo de palabras que cumplen una condición)

Tarea

Desarrollar un algoritmo que lea una oración y determine cuántas palabras tienen una longitud dada.

Ejemplo:

Oración: **UNI solo para emprendedores**

Longitud dada: **4**

Hay 2 palabras de longitud 4

Solución

```
1 #include <iostream>
2 using namespace std;
3 main(){
4     char oracion[100];
5     int conpal=0,i=0, longitud, inicio, fin;
6     cout<<"Ingrese una oracion: ";
7     gets(oracion);
8     cout<<"Ingrese longitud de palabra a contar: ";
9     cin>>longitud;
10    if (oracion[0]!=' '){
11        inicio=0;
12        if (oracion[1]==' ' || oracion[1]=='\0'){
13            fin=0;
14            if ((fin-inicio+1)==longitud)
15                conpal++;
16        }
17    }
18    i=1;
19    while (oracion[i]!='\0'){
20        if (oracion[i]!=' ' && oracion[i-1]==' ')
21            inicio=i;
22        if ((oracion[i+1]==' ' || oracion[i+1]=='\0') && oracion[i]!=' '){
23            fin=i;
24            if ((fin-inicio+1)==longitud)
25                conpal++;
26        }
27        i++;
28    }
29    cout<<"La oracion tiene "<<conpal<<" palabra(s) de longitud "<<longitud<<endl;
30 }
```

Problema sobre cadenas (comparar cadenas)

Tarea

Dada dos palabras
determinar si son iguales o
diferentes

Ejemplo:

palabra1: manzana

Palabra2: naranja

Son diferentes

```
char a[20] = "manzana", b[20] = "naranja";
int i=0;
while (a[i] != '\0') //halla longitud de a
    i++;
int j=0;
while (b[j] != '\0') //halla longitud de b
    j++;
if (i==j){ //compara longitudes
    bool iguales = true;
    for(i=0;i<j;i++)
        if (a[i] != b[i]) //compara caracteres
            iguales=false;
    if (iguales)
        cout << "Son iguales" << endl;
    else
        cout << "Son diferentes" << endl;
}
else
    cout << "Son diferentes" << endl;
```

```
string a = "manzana", b = "naranja";
if (a == b)
    cout << "Son iguales" << endl;
else
    cout << "Son diferentes" << endl;
```

Solución con el tipo **char**

Requiere primero, verificar si las cadenas tienen igual longitud y de ser así, comparar carácter a carácter.

Solución con el tipo **string**

Permite la comparación directa.

Concatenación con el tipo string

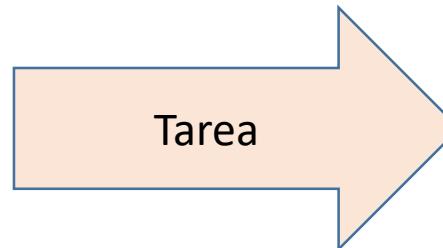
Concatena dos palabras con un espacio en blanco entre ellas

```
string a = "Hola", b = "mundo", c;  
c = a + " " + b;  
cout<<c<<endl; // muestra Hola mundo
```

Construye una cadena a a partir de una vacía y le va agregando caracteres

```
string a="",  
char vocal[5] ={'a','e','i','o','u'};  
for(i=0;i<5;i++)  
    a=a+vocal[i];  
cout<<a<<endl; // muestra aeiou
```

Problema sobre cadenas (comparar cadenas)



Desarrollar un algoritmo que lea una oración y luego una palabra y determine cuántas veces ocurre en la oración.

Ejemplo:

Oración: **Juana es la manzana de la discordia**

Palabra: **la**

la ocurre 2 veces



Detectar el inicio y final de cada palabra, y comparar la palabra ingresada con la que se forma del carácter inicio al carácter fin.