CHUỖI

# Lý thuyết:

Chuỗi trong ngôn ngữ lập trình C thực chất là mảng một chiều của các ký tự mà kết thúc bởi một ký tự **null** '\0'.

Phần khai báo và khởi tạo dưới đây tạo ra một chuỗi bao gồm một từ "Hello". Để giữ các giá trị null tại cuối của mảng, cỡ của mảng các ký tự bao gồm một chuỗi phải nhiều hơn số lượng các ký tự trong từ khóa "Hello".

char loichao[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

Nếu bạn theo quy tắc khởi tạo các chuỗi, bạn có thể viết lệnh như sau:

char loichao[] = "Hello";

Dưới đây là phần biểu diễn ô nhớ cho đoạn chuỗi trên trong ngôn ngữ C/C++:



Thực tế, bạn không đặt ký tự **null** tại vị trí cuối cùng của biến hằng số. Bộ biên dịch C tự động thêm '\0' tại ví trí cuối cùng của chuỗi khi nó khởi tạo chuỗi. Cùng thử ví dụ in ra chuỗi sau đây:

#include <stdio.h>

int main ()

{

char loichao[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

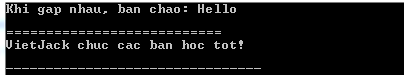
printf("Khi gap nhau, ban chao: %s\n", loichao );

printf("\n===========================\n");

return 0;

}

Biên dịch và chạy chương trình C trên sẽ cho kết quả:



**Các thao tác nhập xuất chuỗi**

Các thao tác nhập/xuất (I/O) chuỗi trong C được thực hiện bằng cách gọi các hàm. Các hàm này là một phần của thư viện nhập/xuất chuẩn tên **stdio.h**. Một chương trình muốn sử dụng các hàm nhập/xuất chuỗi phải có câu lệnh khai báo sau ở đầu chương trình:

#include <stdio.h>;

Sử dụng hàm **gets()** là cách đơn giản nhất để nhập một chuỗi thông qua thiết bị nhập chuẩn. Các ký tự sẽ được nhập vào cho đến khi nhấn phím Enter. Hàm **gets()** thay thế ký tự kết thúc trở về đầu dòng ‘\n’ bằng ký tự ‘\0’. Cú pháp hàm này như sau:

**gets**(str);

Trong đó **str** là một mảng ký tự đã được khai báo.

Tương tự, hàm **puts()** được sử dụng để hiển thị một chuỗi ra thiết bị xuất chuẩn. Ký tự xuống dòng sẽ kết thúc việc xuất chuỗi. Cú pháp hàm như sau:

**puts**(str);

Trong đó **str** là một mảng ký tự đã được khai báo và khởi tạo. Chương trình sau đây nhận vào một tên và hiển thị một thông báo.

**Ví dụ 1:**

#include <stdio.h>

void main()

{

char name[20];

/\* name is declared as a single dimensional character array \*/

puts("Enter your name:"); /\* Displays a message \*/

gets(name); /\* Accepts the input \*/

puts("Hi there: ");

puts(name); /\* Displays the input \*/

}

Nếu tên Lisa được nhập vào, chương trình trên cho ra kết quả:

Enter your name:

Lisa

Hi there:

Lisa

* **Các thao tác Nhập/Xuất chuỗi có định dạng**

Có thể sử đụng các hàm **scanf()** và **printf()** để nhập và hiển thị các giá trị chuỗi. Các hàm này được dùng để nhập và hiển thị các kiểu dữ liệu hỗn hợp trong một câu lệnh duy nhất. Cú pháp để nhập một chuỗi như sau:

**scanf**(“%s”, str);

Trong đó ký hiệu định dạng **%s** cho biết rằng một giá trị chuỗi sẽ được nhập vào. **str** là một mảng ký tự đã được khai báo. Tương tự, để hiển thị chuỗi, cú pháp sẽ là:

**printf**(“%s”, str);

Trong đó ký hiệu định dạng **%s** cho biết rằng một giá trị chuỗi sẽ được hiển thị và **str** là một mảng ký tự đã được khai báo và khởi tạo. Hàm printf() có thể dùng để hiển thị ra các thông báo mà không cần kí tự định dạng.

Có thể sửa đổi chương trình bên trên để nhập vào và hiển thị một tên, sử dụng hàm scanf() và printf().

**Ví dụ 2:**

#include <stdio.h>

void main()

{

char name[20];

/\* name is declared as a single dimensional character

array \*/

printf("Enter your name: "); /\* Displays a message \*/

scanf(“%s”, name); /\* Accepts the input \*/

printf("Hi there: %s", name); /\* Displays the input \*/

}

Nếu nhập vào tên Brendan , chương trình trên cho ra kết quả:

Enter your name: Brendan

Hi there: Brendan

Ngôn ngữ C hỗ trợ một dãy rộng rãi các hàm để thao tác các chuỗi kết thúc là null:

|  |  |
| --- | --- |
| **STT** | **Hàm & Mục đích** |
| 1 | **strcpy(s1, s2);**  Sao chép chuỗi s2 cho chuỗi s1. |
| 2 | **strcat(s1, s2);**  Nối chuỗi s2 vào cuối chuỗi s1. |
| 3 | **strlen(s1);**  Trả về độ dài của chuỗi s1. |
| 4 | **strcmp(s1, s2);**  Trả về 0 nếu s1 và s2 là như nhau; nhỏ hơn 0 nếu s1<s2; lớn hơn 0 nếu s1>s2. |
| 5 | **strstr(s1, s2);**  Trả về con trỏ tới vị trí đầu tiên của chuỗi s2 trong chuỗi s1. |

Dưới đây là ví dụ cho việc sử dụng một vài hàm bên trên:

#include <stdio.h>

#include <string.h> /\* thu vien cho cac ham xu ly chuoi\*/

int main ()

{

char chuoi1[12] = "Hello";

char chuoi2[12] = "Vietjack";

char chuoi3[12];

int dodai ;

/\* sao chep chuoi1 vao trong chuoi3 \*/

strcpy(chuoi3, chuoi1);

printf("Ban su dung ham strcpy( chuoi3, chuoi1) de sao chep: %s\n", chuoi3 );

/\* noi hai chuoi: chuoi1 va chuoi2 \*/

strcat( chuoi1, chuoi2);

printf("Ban su dung ham strcat( chuoi1, chuoi2) de noi chuoi: %s\n", chuoi1 );

/\* tinh do dai cua chuoi1 sau khi noi chuoi \*/

dodai = strlen(chuoi1);

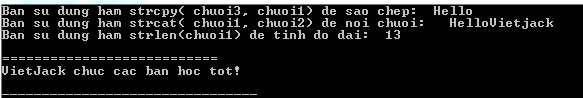
printf("Ban su dung ham strlen(chuoi1) de tinh do dai: %d\n", dodai );

printf("\n===========================\n");

return 0;

}

Biên dịch và chạy chương trình C trên sẽ cho kết quả:



# Bài tập thực hành

**BÀI TẬP 1**

**1. Thời lượng:** Sinh viên thực hành tại nhà

**2. Mô tả bài toán:** Nhập vào 1 chuỗi lien tục không có khoảng trắng. In ra ngược từng ký   
tự của chuỗi như sau:   
Ví dụ:   
Nhập chuỗi: BAOTRAM   
In ra:   
M   
A   
R   
T   
O   
A   
B

**3. Các bước thực hiện:**

Gợi ý: for (ngược từ cuối mảng lên 0)   
n: chiều dài của mảng , dùng hàm strlen(chuoi)

**BÀI TẬP 2**

**1.Thời lượng :** 30 phút

**2.Mô tả bài toán :** Đếm số ký tự trong chuỗi trừ khoảng trắng

**3.Các bước thực hiện :**

for từ đầu chuỗi đến strlen( chuoi)

if(chuoi[i]!=’ ‘) dem++;

**BÀI TẬP 3**

**1.Thời lượng :** 30 phút

**2.Mô tả bài toán :** So sánh chiều dài 2 chuỗi

**3.Các bước thực hiện :**

**BÀI TẬP 4**

**1.Thời lượng :** 30 phút

**2.Mô tả bài toán :** Nhập 1 chuỗi, hãychuyển tất cả ký tự từ Hoa sang thường và ngược lại.

**3.Các bước thực hiện :**

**BÀI TẬP 5**

**1.Thời lượng :** 30 phút

**2.Mô tả bài toán :** Đếm 1 ký tự xuất hiện mấy lần trong chuỗi.

Ví dụ:   
Nhập chuỗi : abcbakfhca   
Nhập ký tự : b   
Ký tự b xuất hiện 2 lần trong chuỗi.

**3.Các bước thực hiện :**

**BÀI TẬP 6**

**1.Thời lượng :** 30 phút

**2.Mô tả bài toán : Sắp xếp các chuỗi**

**3.Các bước thực hiện :**

1. Như chúng ta đã học trong bài lý thuyết, trong C, để sử dụng các hàm về chuỗi, chúng ta cần phải đưa vàohai thư viện chuẩn: **stdio.h**, **string.h**. Câu lệnh sẽ là:

#include <stdio.h>

#include <string.h>

1. Khai báo một mảng ký tự để lưu 5 chuỗi. Câu lệnh sẽ là:

char str\_arr[5][20];

1. Nhập vào 5 chuỗi trong vòng lặp **for**. Câu lệnh sẽ là:

for(i = 0; i < 5; i++)

{

printf("\nEnter string %d: ", i + 1);

scanf("%s", str\_arr[i]);

}

1. So sánh chiều dài của mỗi chuỗi với các chuỗi khác, nếu chiều dài của chuỗi này nhỏ hơn chiều dài của một chuỗi đứng ở vị trí phía sau nó trong mảng, ta sẽ thực hiện đổi chỗ 2 chuỗi đó cho nhau. Câu lệnh sẽ là:

for(i = 0; i < 4; i++)

for(j = i + 1; j < 5; j++)

{

if(strlen(str\_arr[i]) < strlen(str\_arr[j]))

{

strcpy(str, str\_arr[i]);

strcpy(str\_arr[i], str\_arr[j]);

strcpy(str\_arr[j], str);

}

}

Chuỗi **str** được sử dụng trong thao tác hoán đổi hai chuỗi.

1. Hiển thị các chuỗi theo thứ tự đã sắp xếp. Câu lệnh sẽ là:

printf("\nThe strings in descending order of length are:");

for(i = 0; i < 5; i++)

printf("\n%s", str\_arr[i]);

Chúng ta hãy xem chương trình hoàn chỉnh.

1. **Mở chương trình soạn thảo mà bạn dùng để gõ chương trình C.**
2. **Tạo một tập tin mới.**
3. **Gõ vào các dòng lệnh sau đây:**

#include <stdio.h>

#include <string.h>

void main()

{

int i, j;

char str\_arr[5][20], str[20];

for(i = 0; i < 5; i++)

{

printf("\nEnter string %d: ", i + 1);

scanf("%s", str\_arr[i]);

}

for(i = 0; i < 4; i++)

for(j = i + 1; j < 5; j++)

{

if(strlen(str\_arr[i]) < strlen(str\_arr[j]))

{

strcpy(str, str\_arr[i]);

strcpy(str\_arr[i], str\_arr[j]);

strcpy(str\_arr[j], str);

}

}

printf("\nThe strings in descending order of length are:");

for(i = 0; i < 5; i++)

printf("\n%s", str\_arr[i]);

getch();

}

Kết quả của chương trình trên được minh họa như sau:

Enter string 1: This

Enter string 2: sentence

Enter string 3: is

Enter string 4: not

Enter string 5: sorted

The strings in descending order of length are:

sentence

sorted

This

not

is

**BÀI TẬP 7**

**1.Thời lượng :** 30 phút

**2.Mô tả bài toán :**

Nhập vào 4 chuỗi . xuất ra 4 chuỗi

a.Tìm chuỗi có chiều dài ngắn nhất trong mảng chuỗi

b. In ngược từng chuỗi trong mảng chuỗi

**3.Các bước thực hiện :**

**BÀI TẬP 8**

**1.Thời lượng :** 30 phút

**2.Mô tả bài toán :**

In ra menu:   
1.Copy chuoi   
2.Noi chuoi   
3.So sanh chuoi   
4.Tim ky tu trong chuoi   
5.Thoat .

**3.Các bước thực hiện :**

Hướng dẫn :

#include<stdio.h>   
#include<conio.h>   
#include<string.h>   
//khai báo các hàm trên đây   
…………….   
…………….

void main()   
{   
    char st1[50];   
   char st2[50];   
   int c;

   printf("1.Copy chuoi\n");   
   printf("2.Noi chuoi\n");   
   printf("3.So sanh chuoi\n");   
   printf("4.Tim ky tu trong chuoi\n");   
   printf("5.Thoat\n");

 do   
 {

     // Cho  nguoi dung nhap chon (&c)   
      // Dùng switch ….case

  }while(c!=5);

   getch();

}

// để ý hàm nhập chuỗi , trước hàm gets có flushall(), để tránh trường   
hợp gọi hàm nhập chuỗi 2 lần lien tiếp bị lỗi.

void nhapchuoi(char st[])   
{   
    printf("\n nhap chuoi: ");   
    flushall();   
    gets(st);

}

void noichuoi(char st1[],char st2[])   
{  ……….   
     printf("\nchuoi sau khi da ghep %s \n",st1);

}

void copychuoi(char st1[],char st2[])   
{   
   ………….   
   printf("\n chuoi sau khi duoc copy vao la: %s \n",st1);

}

void sosanhchuoi(char st1[],char st2[])   
{   
   int ss=strcmp(st1,st2);   
   if (ss<0)   
        printf……..   
     else if(ss>0)   
           printf……..   
          else   
               printf……..

}

int timkytu(char st[],char kt)   
{   
     Flag=0;   
     ……………   
    Return flag; }