


Trường ĐH CNTP TP. HCM Khoa Công nghệ thông tin Bộ môn Công nghệ phần mềm THỰC HÀNH KỸ THUẬT LẬP TRÌNH	BUỔI 6. XỬ LÝ CHUỖI	
--	----------------------------	---

A. MỤC TIÊU:

- Phân tích các yêu cầu của bài toán.
- Cài đặt được các hàm xử lý chuỗi ký tự cơ bản và sử dụng các hàm trong thư viện **string.h** vào các bài toán.
- Áp dụng các kỹ thuật cài đặt xử lý trên chuỗi vào những bài toán thực tế.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

I. Tóm tắt lý thuyết

1. Khái niệm

Chuỗi ký tự là một dãy các phần tử, mỗi phần tử là một ký tự.

Lưu ý: Chuỗi ký tự được kết thúc bằng ký tự ‘\0’, gọi là ký tự kết thúc chuỗi. Do đó khi khai báo độ dài của chuỗi luôn luôn khai báo dư một phần tử để chứa ký tự ‘\0’.

Ví dụ 1.1. `char s[5] = “CNTT”;` //khai báo chuỗi có 5 phần tử kiểu char và gán dãy ký tự CNTT và chuỗi được lưu trữ như sau:

‘C’	‘N’	‘T’	‘T’	‘\0’
s[0]	s[1]	s[2]	s[3]	s[4]

Lưu ý: Chuỗi rỗng là chuỗi chưa có ký tự nào trong mảng và được ký hiệu “”.

2. Khai báo chuỗi

Để khai báo một chuỗi, ta có 2 cách khai báo sau:

Cách 1: Con trỏ hằng

char <Tên_chuỗi>[<Số ký tự tối đa của chuỗi>;

Ví dụ 2.1. `char hoten[50];`

Cách 2: Con trỏ

char *<Tên_chuỗi>;

Ví dụ 2.2. `char *hoten;`

Ta có thể vừa khai báo vừa gán giá trị cho chuỗi như sau:

Ví dụ 2.3. `char monhoc[50] = “Ky thuat lap trinh”;`

`char s[10] = { ‘K’, ‘T’, ‘L’, ‘T’, ‘\0’};`

Lưu ý: Việc gán nhiều giá trị hằng chuỗi như việc sử dụng dấu ngoặc kép (“”) chỉ hợp lệ khi khởi tạo mảng, tức là lúc khai báo mảng. Các biểu thức trong chương trình như:

```
s = "Hello";
```

```
s[] = "Hello";
```

là không hợp lệ, cả câu lệnh dưới đây cũng vậy:

```
s = {'K', 'T', 'L', 'T', '\0'};
```

Vì vế trái của một lệnh gán chỉ có thể là một phần tử của mảng chứ không thể là cả mảng, chúng ta có thể gán một chuỗi kí tự cho một mảng kiểu char sử dụng một phương pháp như sau:

```
s[0] = 'K';      s[1] = 'T';      s[2] = 'L';      s[3] = 'T';      s[4] = '\0';
```

Nhưng đây không phải là một phương pháp thực tế. Để gán giá trị cho một chuỗi kí tự, chúng ta có thể sử dụng các hàm kiểu **strcpy**, hàm này được định nghĩa trong thư viện **string.h** và được giải thích cụ thể trong phần 3.

3. Các thao tác xử lý chuỗi

a. Nhập dữ liệu cho chuỗi

Sử dụng hàm `gets()` (thuộc thư viện `<stdio.h>`) được coi là cách thức đơn giản và phổ biến nhất để ta tiến hành nhập liệu cho chuỗi. Cú pháp, sử dụng hàm này như sau: `gets(Tên_chuỗi);`

Ví dụ 3.1. Nhập dữ liệu từ bàn phím cho chuỗi `s` có kích thước tối đa 30 ký tự:

```
char s[30];  
printf("\n Moi ban nhap mot chuoi: ");  
gets(s);
```

Ngoài ra, ta cũng có thể sử dụng hàm `scanf()` với chuỗi định dạng `“%s”` để nhập dữ liệu cho chuỗi. Tuy nhiên, cách nhập này không cho phép nhập chuỗi có chứa khoảng trắng.

b. Xuất dữ liệu cho chuỗi

Có 2 cách xuất chuỗi ra màn hình như sau:

Cách 1: Sử dụng hàm `printf` với đặc tả `“%s”`

```
char monhoc[50] = "Tin hoc co so A";  
printf("%s", monhoc);    //Không xuống dòng
```

Cách 2: Sử dụng hàm `puts`

```
char monhoc[50] = "Tin hoc co so A";  
puts(monhoc);    //Tự động xuống dòng
```

c. Các hàm xử lý chuỗi trong thư viện <string.h>

STT	TÊN HÀM	CHỨC NĂNG	VÍ DỤ
1	int strlen (char s[]);	Trả về độ dài của chuỗi s.	char *s = "Borland International"; printf("Do dài s: %d", strlen(s)); Kết quả: Do dài s: 21
2	strcpy (char dest[], char src[]);	Sao chép nội dung chuỗi src vào chuỗi dest.	char dest[10]; char *src = "abcdefghi"; strcpy(dest, src); printf("%s\n", dest); Kết quả: abcdefghi
3	strncpy (char dest[], char src[], int n);	Chép n ký tự từ chuỗi src sang chuỗi dest. Nếu chiều dài src < n thì hàm sẽ điền khoảng trắng cho đủ n ký tự vào dest.	char dest[4]; char *src = "abcdefghi"; strncpy(dest, src, 3); printf("%s\n", dest); Kết quả: abc
4	strcat (char s1[], char s2[]);	Nối chuỗi s2 vào chuỗi s1.	char *s1 = "Khoa"; char *s2 = "CNTT"; strcat(s1, s2); printf("%s\n", s1); Kết quả: Khoa CNTT
5	strncat (char s1[], char s2[], int n)	Nối n ký tự đầu tiên của chuỗi s2 vào chuỗi s1.	char *s1 = "Khoa"; char *s2 = "CNTT"; strncat(s1, s2, 2); printf("%s\n", s1); Kết quả: Khoa CN
6	int strcmp (char s1[], char s2[])	So sánh 2 chuỗi s1 và s2 theo nguyên tắc thứ tự từ điển. Phân biệt chữ hoa và thường. Trả về: • 0: nếu s1 bằng s2. • >0: nếu s1 lớn hơn s2. • <0: nếu s1 nhỏ hơn s2.	char *s1 = "abcd"; char *s2 = "abCD"; if(strcmp(s1, s2)==0) printf("Giống nhau"); else printf("Khác nhau"); Kết quả: Khác nhau
7	int strncmp (char s1[], char s2[], int n)	Tương tự như strcmp(), nhưng chỉ so sánh n ký tự đầu tiên của hai chuỗi.	char *s1 = "abcd"; char *s2 = "abef"; if(strncmp(s1, s2, 2)==0) printf("Giống nhau"); else printf("Khác nhau"); Kết quả: Giống nhau

STT	TÊN HÀM	CHỨC NĂNG	VÍ DỤ
8	int stricmp (char s1[], char s2[])	Tương tự như strcmp(), nhưng không phân biệt hoa thường.	char *s1 = "abcd"; char *s2 = "abCD"; if(stricmp(s1, s2)==0) printf("Giống nhau"); else printf("Khác nhau"); Kết quả: Giống nhau
9	int strnicmp (char s1[], char s2[], int n);	Tương tự như stricmp(), nhưng chỉ so sánh n ký tự đầu tiên của hai chuỗi.	char *s1 = "aBcd"; char *s2 = "Abef"; if(strnicmp(s1, s2, 2)==0) printf("Giống nhau"); else printf("Khác nhau"); Kết quả: Giống nhau
10	char *strchr (char s[], char c);	Tìm lần xuất hiện đầu tiên của ký tự c trong chuỗi s. Trả về: <ul style="list-style-type: none"> • NULL: nếu không có. • Địa chỉ c: nếu tìm thấy. 	char s[15]; char *ptr, c = 'm'; strcpy(s, "Vi du tim ky tu"); ptr = strchr(s, c); if (ptr) printf("Ky tu %c tai: %d", c, ptr-s); else printf("Khong tim thay"); Kết quả: Ky tu m tai: 8

II. Bài tập có hướng dẫn

Bài 4. Nhập vào một chuỗi từ bàn phím. Tính và xuất độ dài của chuỗi (không dùng hàm trong thư viện string.h).

Hướng dẫn:

Phân tích:

- Input: chuỗi s.
- Output: n (là chiều dài chuỗi s).

Chương trình mẫu:

```
#include <stdio.h>
#include <conio.h>
int length(char s[])
{
    int i = 0;
    while(s[i] != '\0')
        i++;
}
```

```

        return i;
    }
    void main()
    {
        //sinh viên khai báo chuỗi, gọi và in giá trị cho hàm length
        getch();
    }

```

Bài 5. Nhập vào 1 chuỗi và 1 ký tự c, kiểm tra ký tự c có trong chuỗi hay không, nếu có đưa ra số lần xuất hiện của ký tự đó trong chuỗi.

Hướng dẫn:

Phân tích:

- Input: chuỗi s và ký tự c.
- Output: Trả lời ký tự c có trong chuỗi s hay không.

Chương trình mẫu:

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
int kiemTra(char s[], char c) {
    int dem = 0;
    //for(int i = 0; i < strlen(s); i++) {
    for(int i = 0; s[i] != '\0'; i++) {
        if(s[i] == c)
            dem++;
    }
    return dem;
}
void main() {
    /*sinh viên khai báo chuỗi và 1 ký tự, gọi và in giá trị cho hàm kiemtra*/
    getch();
}

```

Bài 6. Nhập vào một chuỗi từ bàn phím. Đếm có bao nhiêu từ trong chuỗi vừa nhập (giả sử chuỗi nhập vào đúng chuẩn).

Hướng dẫn:

Phân tích:

- Input: chuỗi s.
- Output: n (là số từ của chuỗi s).

Chương trình mẫu:

```

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

```

```

int demTu(char s[]) {
    int dem = 0;
    //for(int i = 0; i < strlen(s); i++) {
    for(int i = 0; s[i] != '\0'; i++) {
        if(s[i] == ' ')
            dem++;
    }
    return dem;
}

void main() {
    //sinh viên tự viết code gọi hàm và xuất kết quả
    getch();
}

```

III. Bài thực hành trên lớp

Bài 7. Kiểm tra xem chuỗi s có chứa toàn ký số hay không?

Bài 8. Viết hàm đổi những kí tự đầu tiên của mỗi từ thành chữ in hoa và những từ không phải đầu câu sang chữ thường.

Bài 9. Viết hàm xóa những khoảng trắng thừa trong chuỗi (hay còn gọi là chuẩn hóa chuỗi).

Bài 10. Viết hàm tìm kiếm tên trong một chuỗi họ tên. Nếu có thì xuất ra là tên chuỗi họ tên, ngược lại thông báo tên không tồn tại.

Bài 11. Viết hàm cắt chuỗi họ tên thành 2 chuỗi con: họ lót và tên.

Ví dụ: chuỗi họ tên là: “Nguyễn Văn Anh” cắt ra 2 chuỗi là chuỗi họ lót: “Nguyễn Văn”, chuỗi tên là: “Anh”.

Bài 12. Nhập vào một danh sách sinh viên và hiển thị danh sách sinh viên ra màn hình.

Áp dụng giải thuật Brute Force, tìm và xuất vị trí các chuỗi P trong T,

- a) Với T là chuỗi nhập vào từ bàn phím.
- b) Với T là chuỗi văn bản đọc từ file text.

IV. Bài thực hành về nhà

Bài 13. Kiểm tra xem chuỗi nhập vào có đối xứng hay không?

Bài 14. Nhập vào chuỗi s1 và s2, cho biết vị trí xuất hiện của chuỗi s2 trong s1 (nếu có). Nếu s2 không có trong s1, thực hiện nối s2 vào cuối s1.

Bài 15. Nhập vào chuỗi str, chuỗi cần chèn strInsert và vị trí cần chèn vt. Hãy chèn chuỗi strInsert vào chuỗi str tại vị trí vt.

Bài 16. Nhập một chuỗi bất kì, yêu cầu nhập 1 kí tự muốn xóa. Thực hiện xóa tất cả những kí tự đó trong chuỗi.

Bài 17. Nhập vào 1 mảng các chuỗi. Tìm xuất ra màn hình những chuỗi chứa toàn ký tự số.

Bài 18. Cho 2 chuỗi s1 và s2 là 2 từ (chỉ chứa chữ cái hoặc ký số). Tính xem cần thực hiện bao nhiêu phép biến đổi để chuyển chuỗi s1 thành s2. Biết rằng có 3 phép biến đổi

- Thêm ký tự
- Xóa ký tự
- Đổi ký tự

Ví dụ: xét 2 từ s1= “Kitten” và s2=“Sitting”

Cần thực hiện 3 phép đổi để chuyển s1 thành s2. Đó là:

- Đổi ‘K’ thành ‘S’
- Đổi ‘e’ thành ‘i’
- Thêm ‘g’ vào cuối.

---HẾT---