

CHƯƠNG 5 CHUỖI KÝ TỰ

Chuỗi ký tự là trường hợp đặc biệt của mảng một chiều. Chương này mô tả một số hàm thư viện thao tác trên chuỗi và các kỹ thuật cài đặt xử lý trên chuỗi.

I. TÓM TẮT LÝ THUYẾT

I.1. Khái niệm

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

Lưu ý: Chuỗi ký tự được kết thúc bằng ký tự '\0'. Do đó khi khai báo độ dài của chuỗi luôn luôn khai báo dư 1 phần tử để chứa ký tự '\0'.

Ví dụ: `char S[5]="CNTT"` //khai báo chuỗi có 5 phần tử kiểu char và gán dãy ký tự CNTT và chuỗi.

C	N	T	T	\0
Phần tử S[0]	Phần tử S[1]	Phần tử S[2]	Phần tử S[3]	Phần tử S[4]

Chuỗi rỗng là chuỗi chưa có ký tự nào trong mảng ký hiệu ""

I.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: `char chuoi[25];`

Ý nghĩa khai báo **1 mảng kiểu ký tự tên là chuoi** có 25 phần tử (như vậy tối đa ta có thể nhập 24 ký tự vì **phần tử thứ 25 đã chứa ký tự kết thúc chuỗi '\0'**)

❖ Cách 2: Con trỏ

`char *< Tên chuỗi >;`

Ví dụ : `char *chuoi;`

I.3. Các thao tác trên chuỗi

a. *Nhập chuỗi*

Cú pháp : `char *gets(char *s);`

Nhận các ký tự nhập từ phím cho đến khi nhấn phím Enter và đưa vào s.

Ví dụ:

```
void main()
{
    char chuoi[80];
    printf("Nhap vao chuoi:");
    gets(chuoi);
    printf("Chuoi vua nhap la: %s\n", chuoi);
}
```

b. Xuất chuỗi

Cú pháp : int puts(const char *s);

Xuất chuỗi s ra màn hình.

Ví dụ:

```
void main()
{
    char chuoi[] = "Vi du xuat chuoi\n";
    puts(string);
}
```

c. Các hàm thư viện (string.h)

STT	TÊN HÀM	CHỨC NĂNG	VÍ DỤ
1	int strlen(char s[]);	Trả về độ dài của chuỗi s.	<pre>char *s = "Borland International"; printf("Do dai s: %d\n", strlen(s));</pre> <p>Kết quả: Do dai s: 21</p>
2	strcpy(char dest[], char src[]);	Sao chép nội dung chuỗi src vào chuỗi dest.	<pre>char dest[10]; char *src = "abcdefghi"; strcpy(dest, src); printf("%s\n", dest);</pre> <p>Kết quả: abcdefghi</p>
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.	<pre>char dest[4]; char *src = "abcdefghi"; strncpy(dest, src, 3); printf("%s\n", dest);</pre> <p>Kết quả: abc</p>
4	strcat(char s1[], char s2[]);	Nối chuỗi s2 vào chuỗi s1.	<pre>char *s1 = "Khoa "; char *s2 = "CNTT"; strcat(s1, s2); printf("%s\n", s1);</pre> <p>Kết quả: Khoa CNTT</p>
5	strncat(char s1[], char s2[], int n)	Nối n ký tự đầu tiên của chuỗi s2 vào chuỗi s1.	<pre>char *s1 = "Khoa "; char *s2 = "CNTT"; strncat(s1, s2, 2); printf("%s\n", s1);</pre> <p>Kết quả: Khoa CN</p>
6	Int strcmp(char s1[], char s2[])	So sánh 2 chuỗi s1 và s2 theo nguyên tắc <u>thứ tự từ điển</u> . Phân biệt chữ hoa và thường.	<pre>char *s1 = "abcd"; char *s2 = "abCD"; if(strcmp(s1, s2)==0)</pre>

		Trả về: <ul style="list-style-type: none"> • 0 : nếu s1 bằng s2. • >0: nếu s1 lớn hơn s2. • <0: nếu s1 nhỏ hơn s2. 	<pre>printf("Giống nhau"); else printf("Khác nhau");</pre> Kết quả: Khác nhau
7	int strcmp(char s1[], char s2[], int n)	Tương tự như strcmp(), nhưng chỉ so sánh n ký tự đầu tiên của hai chuỗi.	<pre>char *s1 = "abcd"; char *s2 = "abef"; if(strcmp(s1, s2, 2)==0) printf("Giống nhau"); else printf("Khác nhau");</pre> Kết quả: Giống nhau
6	int stricmp(char s1[], char s2[])	Tương tự như strcmp(), nhưng không phân biệt hoa thường.	<pre>char *s1 = "abcd"; char *s2 = "abCD"; if(stricmp(s1, s2)==0) printf("Giống nhau"); else printf("Khác nhau");</pre> Kết quả: Giống nhau
7	int strnicmp(char s1[], char s2[], int n);	Tương tự như stricmp(), nhưng chỉ so sánh n ký tự đầu tiên của hai chuỗi.	<pre>char *s1 = "aBcd"; char *s2 = "Abef"; if(strnicmp(s1, s2, 2)==0) printf("Giống nhau"); else printf("Khác nhau");</pre> Kết quả: Giống nhau
8	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. 	<pre>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");</pre> Kết quả: Ky tu m tai: 8
9	char *strstr(char s1[], char s2[]);	Tìm sự xuất hiện đầu tiên của chuỗi s2 trong chuỗi s1. Trả về: <ul style="list-style-type: none"> • NULL: nếu không có. • Ngược lại: Địa chỉ bắt đầu chuỗi s2 trong s1. 	<pre>char *s1 = "Borland International"; char *s2 = "nation", *ptr; ptr = strstr(s1, s2); printf("Chuoi con: %s\n", ptr);</pre> Kết quả: Chuoi con: national
10	char *strtok(char s1[], char s2[]);	<ul style="list-style-type: none"> • Nếu s2 có xuất hiện trong s1: Tách chuỗi s1 thành hai chuỗi: Chuỗi đầu là những ký tự cho đến khi gặp chuỗi s2 đầu tiên, chuỗi sau là những ký tự còn lại của s1 sau khi đã bỏ đi chuỗi s2 xuất hiện trong s1. <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">S1:</div> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <div style="display: flex; justify-content: space-between; width: 100%;"> <div style="width: 40%;"></div> <div style="width: 20%; text-align: center;">S2</div> <div style="width: 40%;"></div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px; width: 30%;">S1'</div> <div style="border: 1px solid black; padding: 2px; width: 30%;">S1''</div> </div> </div> </div>	<pre>char input[16] = "abc,d"; char *p; // Lay chuoi dau p = strtok(input, ","); if (p) printf("S11: %s\n", p); /*Lay chuoi con lai, tham so dau la NULL*/ p = strtok(NULL, ","); if (p) printf("S12: %s\n", p);</pre> Kết quả: S11: abc S12: d

		<ul style="list-style-type: none"> Nếu s2 không xuất hiện trong s1 thì kết quả chuỗi tách vẫn là s1. 	
<p>🔗 Lưu ý: Cách truy xuất các ký tự tương tự như mảng một chiều.</p>			

d. Ví dụ

Nhập vào một chuỗi ký tự, xuất ra màn hình chuỗi bị đảo ngược thứ tự các ký tự.

Ví dụ: Nhập vào: **Tran minh thai**. Xuất ra màn hình: **iaht hnim narT**

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

void DaoChuoi(char *s1, char *s2)
{
    int l=strlen(s1);
    for(int i=0; i<l; i++)
        s2[i]=s1[l-i-1];

    s2[i]='\0';
}

void main()
{
    char *s1, *s2;
    clrscr();
    printf("\nNhap vao chuoi ky tu: ");
    gets(s1);
    DaoChuoi(s1, s2);
    printf("\nKet qua sau khi dao nguoc chuoi: %s", s2);
}
```

II. BÀI TẬP

II.1. Bài tập cơ bản

- Cho biết kết quả của đoạn chương trình sau:

```
char input[20]="Truong cao dang CNTT", *p, *temp;
strcpy(temp, input);
do
{
    p = strtok(temp, " ");
    printf("%s\n",p);
```

```
p = strtok(NULL, "");
strcpy(temp, p);
}while(p!=NULL);
printf("Chuoi temp: %s \n Chuoi input: %s", temp, input);
```

2. Cho biết kết quả của đoạn chương trình sau:

```
char s1[20]="Truong cao dang CNTT", s1[10]="Tp. HCM", *input, *s3;
strcpy(input, s1);    strcpy(s3,"aeiou"); strcat(input, s2);
int n=strlen(input), k=0;
printf("Chuoi: %s",input);
for(int i=0; i<n; i++)
{
    if(strchr(s3, input[i]))
        k++;
}
printf("\nKet qua: %d", k);
```

3. Viết chương trình nhập vào một chuỗi ký tự, đếm số ký tự có trong chuỗi.
4. Viết chương trình đếm có bao nhiêu khoảng trắng trong chuỗi.
5. Viết chương trình nhập vào một chuỗi, hãy loại bỏ những khoảng trắng thừa trong chuỗi.
6. Viết chương trình nhập vào hai chuỗi s1 và s2, nối chuỗi s2 vào s1. Xuất chuỗi s1 ra màn hình.
7. Đổi tất cả các ký tự có trong chuỗi thành chữ thường (không dùng hàm `strlwr`).
8. Đổi tất cả các ký tự trong chuỗi sang chữ in hoa (không dùng hàm `struppr`).
9. Viết chương trình đổi những ký tự đầu tiên của mỗi từ thành chữ in hoa.
10. Viết chương trình đổi chữ xen kẽ 1 chữ hoa và 1 chữ thường.
Ví dụ: nhập *ABCDEfgh* đổi thành *AbCdEfGh*
11. Viết chương trình đảo ngược các ký tự trong chuỗi .
Ví dụ: nhập *ABCDE*, xuất ra màn hình là: *EDCBA*
12. Viết chương trình tìm kiếm 1 ký tự xem có trong chuỗi hay không, nếu có xuất ra vị trí của từ đó.
13. Viết 1 chương trình đếm một ký tự xuất hiện bao nhiêu lần trong chuỗi.

14. Viết chương trình tìm kiếm tên trong chuỗi họ tên. Nếu có thì xuất ra là tên này đã nhập đúng, ngược lại thông báo là đã nhập sai.
15. Viết chương đảo vị trí của từ đầu và từ cuối.
Ví dụ: nhập “bo an co” xuất ra “co an bo”
16. Viết hàm cắt chuỗi họ tên thành chuỗi họ lót và chuỗi tên.
Ví dụ: chuỗi họ tên là: “Nguyễn Văn A” cắt ra 2 chuỗi là chuỗi họ lót: “Nguyễn Văn”, chuỗi tên là: “A”
17. Nhập một chuỗi bất kỳ, sau đó hỏi người dùng cần tách bắt đầu từ đâu trong chuỗi trở về sau.
Ví dụ: Nhập chuỗi S1: “trường Cao Đẳng Công Nghệ Thông tin”. Người nhập muốn tách bắt đầu từ chữ “Công” thì sẽ xuất ra chuỗi “Công Nghệ Thông Tin” ra màn hình.
18. Viết hàm kiểm tra xem chuỗi có đối xứng hay không?.
19. Viết hàm tra xem trong chuỗi có ký tự số hay không nếu có tách ra thành một mảng số riêng.
20. Nhập một chuỗi bất kì, yêu cầu nhập 1 ký tự muốn xóa. Thực hiện xóa tất cả những ký tự đó trong chuỗi.
21. Viết chương trình tìm kiếm xem ký tự nào xuất hiện nhiều nhất trong chuỗi.
22. Viết 1 chương trình xóa một từ nào đó trong chuỗi.
Ví dụ: Chuỗi ban đầu: “CAO DANG CNTT”
Nhập: “CNTT”, và kết quả xuất ra: “CAO DANG”

II.2. Bài tập luyện tập và nâng cao

23. Đổi các từ ở đầu câu sang chữ hoa và những từ không phải đầu câu sang chữ thường.
Ví dụ: nGuYen vAN a đổi thành: Nguyễn Văn A
24. (*) Viết chương trình đảo ngược thứ tự các từ có trong chuỗi
Ví dụ: Nhập Truong CD CNTT TpHCM
Xuất ra màn hình là: TpHCM CNTT CD Truong
25. Nhập 1 chuỗi bất kì, liệt kê xem mỗi ký tự xuất hiện mấy lần.
26. Viết hàm kiểm tra xem trong 2 chuỗi có bao nhiêu ký tự giống nhau.
27. Viết chương trình mình chạy từ trái qua phải màn hình.

28. Viết 1 chương trình chèn 1 từ ở bất cứ vị trí nào mà người dùng yêu cầu.
29. (*) Viết chương trình nhập vào một chuỗi đếm xem chuỗi có bao nhiêu từ. Các từ cách nhau bằng khoảng trắng, dấu chấm câu: dấu chấm (.), dấu phẩy (,), dấu chấm phẩy (;), dấu hỏi (?) và dấu chấm than (!).
30. (**) Viết chương trình hiển thị một chuỗi ký tự. Chương trình cho phép di chuyển dấu nháy sang trái, sang phải, lên dòng hay xuống dòng bằng phím mũi tên, chèn hay xoá ký tự tại vị trí dấu nháy.

III. KẾT LUẬN

- ❖ Cũng giống như kiểu mảng một chiều, thao tác truy xuất các phần tử trên chuỗi hoàn toàn tương tự. Bên cạnh đó, kiểu dữ liệu này còn được cài đặt sẵn một số hàm thư viện rất hữu ích nên trong quá trình thao tác trên chuỗi nên khi cài đặt ta cố gắng **tận dụng tối đa những hàm liên quan**.
- ❖ **Không nên sử dụng hàm scanf()** để nhập chuỗi trong trường hợp chuỗi dữ liệu nhập vào có chứa khoảng trắng.
- ❖ Nếu **nhập chuỗi phía sau hàm scanf()** nên chèn hàm **fflush(stdin)** hoặc hàm **flushall()** giữa **scanf** và **gets()** để xóa vùng đệm, tránh trường hợp chương trình **bỏ qua hàm gets()** do trong vùng đệm còn lưu ký tự xuống dòng của phím **ENTER**.
- ❖ Khi thao tác trên chuỗi lưu ý phải **đảm bảo chuỗi được kết thúc bằng ký tự kết thúc '\0'**.