Nguyễn Quang Ngọc

Chương 4. Chuỗi

4.1 Chuỗi :

1. Khai báo :

char S[n];

- n là hằng nguyên ≥ 1;

2. Gán giá trị hằng chuỗi cho biến chuỗi:

S = “$a_0a_1. . .a_m$”

- **$m \leq n - 2$** ,
- Với các a_i là các ký tự trên bàn phím,
- a_i là ký tự thứ i trong chuỗi,
- S[i] = ‘a_i’.

Ví dụ :

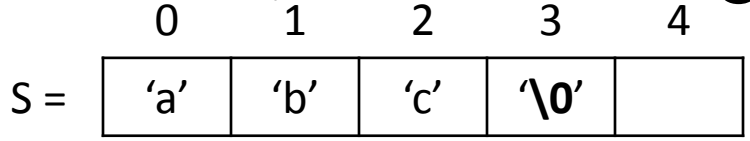
char S[5];

S = “abc”

Ta có :

a_0=’a’ , a_1=’b’, a_2=’c’

- Chuỗi “abc” được lưu trữ trong S như sau :



- S cũng có thể được gán như sau :

char S[5] = {'a', 'b', 'c', '\0'};

Hay char S[5] = {97, 98, 99, '\0'};

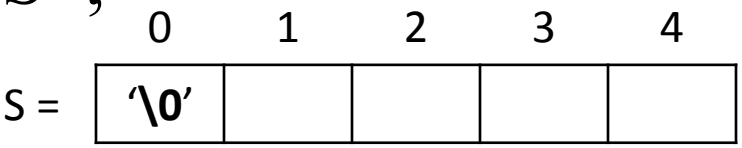
(97 là mã ASCII của ký tự 'a‘, 98 là mã ASCII của ký tự ‘b’, 99 là mã ASCII của ký tự ‘c’)

Ví dụ :

char S[5];

S = ”” ;

Gán chuỗi rỗng cho S ,



- S cũng có thể được gán như sau :

char S[5] = {'\0'};

Hay

char S[5];

S[0] = 0 ;

4.2 Hàm : #include <string.h>

1) **strlen(S)** : số ký tự của S.

VD : char S[5] = ”abc” ;

printf(“%d”, strlen(S)); → **3**

2) **strcpy(S1, S2)** : copy S2 cho S1

VD : char S1[5], S2[5] = "abc“ ;

strcpy(S1, S2); → S1 = "abc"

- Lệnh gán sau là sai :

```
char S1[5], S2[5] = "abc“ ;
S1 = S2;  // SAI
```

- Các lệnh gán sau là sai :

1) char S1[5], S2[5] = "abc“ ;

S1 = S2; // SAI vì vế phải là BIẾN

2) char S1[5] ;

S1 = “abc”; // SAI vì vế phải chỉ có 3 ký tự

3) **strncpy(S1, S2, n)** : copy n ký tự đầu tiên của S2 cho S1

VD 1 : char S1[15], S2[10] = "abcdef";

strcpy(S1, S2, 3); → S1 = "abc"

VD 2 : char S1[15], S2[10] = "abc";

strcpy(S1, S2, 4); → S1 = "abc"

4) **strcat(S1, S2)** : ghép S2 vào cuối S1.

VD :

char S1[15]="abc" , S2[10]="de";

strcat(S1, S2); → "abcde"

4) **strncat(S1, S2, n)** : ghép n ký tự đầu tiên của S2 vào cuối S1.

VD :

char S1[15]="abc" , S2[10]="defghij";

strncat(S1, S2, 2); → "abcde"

5) **strcmp(S1, S2)** : cho giá trị 0 nếu S1 bằng S2, cho giá trị khác không nếu S1 khác S2.

VD 1 :

char S1[15]="abc" , S2[10]=“abc";

B= strcmp(S1, S2); $\rightarrow$ B = 0

VD 2 :

char S1[15]="abc" , S2[10]=“aBc";

B= strcmp(S1, S2); $\rightarrow$ B $\neq$ 0

VD 3 :

char S1[15]="abc" , S2[10]=“aefd";

B= strcmp(S1, S2); $\rightarrow$ B $\neq$ 0

6) **strcmpi(S1, S2)** : như strcmp nhưng không phân biết chữ in hay chữ thương khi so sánh.

VD 1 :

char S1[15]="abc" , S2[10]=“abc";

B= strcmp(S1, S2); → B = 0

VD 2 :

char S1[15]="a**b**c" , S2[10]=“a**B**c";

B= strcmp(S1, S2); → B = 0

7) **strchr(S, c)** : cho giá trị là một phần của chuỗi S bắt đầu từ ký tự **c** *đầu tiên* trong S cho đến cuối chuỗi S.

VD :

```
char S1[20],S2[20];
strcpy(S1,"abcidefghilk");
strcpy(S2, strchr(S1,'i'));  (1)
```

(1) → S2 là "**i**defgh**i**lk"

- Nếu không có ký tự **c** trong **S** thì hàm **strchr(S, c)** cho giá trị là **NULL**. Thay *(1)* ta thực hiện

```
if (strchr(S1,c)!=NULL)
        strcpy(S2, strchr(S1,c));
```

8) **strrchr(S, c)** : cho giá trị là một phần của chuỗi S bắt đầu từ ký tự **c** *cuối cùng* trong S cho đến cuối chuỗi S.

VD :

```
char S1[20],S2[20];
int vt;
strcpy(S1,"abcidefghilk");
strcpy(S2, strchr(S1,'i'));
```

→ S2 là "**i**lk"

- Nếu không có ký tự **c** trong **S** thì hàm **strrchr(S, c)** cho giá trị là **NULL**.

9) **strstr(S1, S2)** : cho giá trị là một phần của chuỗi S1 bắt đầu từ chuỗi S2 *đầu tiên* trong S1 cho đến cuối chuỗi S1.

VD :

```
char S1[20],S2[20];
strcpy(S1,"abcdiefghielk");
strcpy(S2, strstr(S1,"ie"));
```

→ S2 = "**ie**fgh**ie**lk"

- Nếu không có **S2** trong **S1** thì hàm **strstr(S1, S2)** cho giá trị là **NULL**.

10) **strlwr(S)** : cho giá trị là chuỗi S trong đó các ký tự alphabet IN sẽ được thay bằng các ký tự alphabet thường, các ký tự khác giữ nguyên.

VD :

```
char S1[20],S2[20];
strcpy(S1, "A+B=C");
strcpy(S2, strlwr(S1));
```
→ S2 = "a+b=c“

strupr(S) : cho giá trị là chuỗi S trong đó các ký tự alphabet thường sẽ được thay bằng các ký tự alphabet IN, các ký tự khác giữ nguyên.

11) **strrev (S)** : cho giá trị là chuỗi đảo ngược của chuỗi S.

VD :

```
char S1[20],S2[20];

strcpy(S1,"123");
strcpy(S2, strrev(S1));   → S2 = "321"
```

12) **strtok (S)** (soạn ở chương con trỏ):

4.3 Mảng chuỗi :

Khai báo :

```
char  M[m][n];
```

Các chuỗi : M[0], M[1], . . ., M[m-1], mỗi chuỗi có nhiều nhất n-2 ký tự.

VD :

```
char M[3][5];
int i;

strcpy(M[0], ″ab″);
strcpy(M[1], ″abc″);
strcpy(M[2], ″abcd″);
for(i=0 ; i < 3; i=i+1)
        printf(″%s\n ″, M[i]);
```

VD :

```
char M[3][5]={“ab”, “abc”, “abcd”};
int i;

for(i=0 ; i < 3; i=i+1)
printf(″%s\n ″, M[i]);
```

VD :

```
char M[3][5]={{'a','b','\0'}, {'a','b','c','\0'}, {'a','b','c','d','\0'}};
int i;
for(i=0 ; i < 3; i=i+1)
printf("%s\n", M[i]);
```

VD :

```
char M[3][5]={{97, 98 , 0}, {'a','b','c','\0'}, {'a','b','c','d','\0'}};
int i;
for(i=0 ; i < 3; i=i+1)
printf("%s\n", M[i]);
```

4.4 Nhập, viết chuỗi :

4.4.1 Nhập chuỗi :

1. scanf( ) :

VD 1 :

```
char S[20];
scanf("%s", S);
```

Giả sử chuỗi nhập là **toi** , S có giá trị là "**toi**" .

VD 2 :

```
char S[20];
scanf("%s", S);
```

Giả sử chuỗi nhập là **toi□□di□hoc** , S có giá trị là "**toi**" .

VD 3 :

```
char S[20];
scanf("%s", S);
```

Giả sử chuỗi nhập là □□□**toi**□□**di**□**hoc** , S có giá trị là "**toi**" .

VD 4 :

```
char S1[20], S2[20];
scanf("%s%s", S1,S2);
```

Giả sử chuỗi nhập là □□□**toi**□□**di**□**hoc** , S1 có giá trị là "**toi**“, S2 có giá trị là “**di**" .

VD 5 :

```
char S1[20], S2[20];
scanf("%s%s", S1); (1)
scanf("%s%s", S2);
```

Giả sử chuỗi nhập ở (1) là □□□**toi**□□**di**□**hoc** , S1 có giá trị là "**toi**“, S2 có giá trị là “**di**" .

2. gets( ) : nhấp tất cả các ký tự cho biến chuỗi

VD :

```
char S[20];
gets(S);
```

Giả sử chuỗi nhập là □**toi**□□**di**□**hoc** , S có giá trị là □**toi**□□**di**□**hoc**.

4.4.2 Viết chuỗi :

1. printf( ) :

VD :

```
char S[20];
strcpy(S, "toi□□di□hoc");
printf("%s", S);
```

Trên màn hình : **toi□□di□hoc**

2. puts( ) :

VD :

```
char S[20];
strcpy(S, "toi□□di□hoc");
puts(S);
```

Trên màn hình : **toi□□di□hoc**

- gets , puts thuộc <stdio.h> .