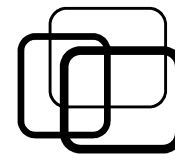


Chuỗi ký tự

GV. Nguyễn Minh Huy

Nội dung



- Chuỗi ký tự.
- Thư viện xử lý chuỗi.



- **Chuỗi ký tự.**
- Thư viện xử lý chuỗi.

Chuỗi ký tự



■ Chuỗi ký tự trong C:

- Mảng ký tự + **phần tử cuối '\0'**.

→ Chiều dài chuỗi = số phần tử mảng – 1;

- Khai báo chuỗi:

char <Tên chuỗi> [<Chiều dài chuỗi> + 1];

char **s1**[5]; **s1**

0	1	2	3	4
?	?	?	?	\0

- Khởi tạo chuỗi:

char <Tên chuỗi> [] = “<**Chuỗi khởi tạo**>”;

char **s2**[] = “Hello”; **s2**

0	1	2	3	4	5
H	e	l	l	o	\0

 // Khởi tạo chuỗi



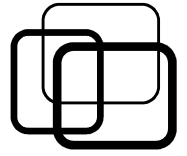
■ Thao tác chuỗi:

■ Nhập chuỗi:

- `scanf("%s[^\n]", chuoi).`
- `fgets(chuoi, MAX, stdin).`
- `cin.getline(chuoi, MAX) (C++).`

■ Xuất chuỗi:

- `printf("%s", chuoi).`
- `puts(chuoi).`
- `cout << chuoi (C++).`



■ Thao tác chuỗi:

■ Đếm chiều dài:

- Phần tử cuối `'\0'`.

■ Đếm từ:

- Khoảng trắng: `' '`, `'\n'`, `'\t'`, `'\r'`.
- **isspace**(<ký tự>) (thư viện **<ctype.h>**).

■ Viết hoa chữ cái đầu mỗi từ:

- Chữ thường: `'a' - 'z'`.
- **islower**(<ký tự>) (thư viện **<ctype.h>**).
- Viết hoa: ký tự - 32.
- **toupper**(<ký tự>) (thư viện **<ctype.h>**).

Nội dung



- Chuỗi ký tự.
- **Thư viện xử lý chuỗi.**



■ Thư viện <string.h>:

- `strlen(<chuoi>)`: đếm chiều dài chuỗi.

```
char s1[ ] = "Hello World";  
char s2[ ] = "Hello World\n";  
char s3[ ] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd', '\n', '\0' };
```

```
int chieuDai1 = strlen(s1); // chieuDai1 = 11  
int chieuDai2 = strlen(s2); // chieuDai2 = 12  
int chieuDai3 = strlen(s3); // chieuDai3 = 12
```



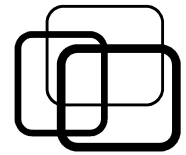

■ Thư viện <string.h>:

■ strcpy(<chuỗi đích>, <chuỗi nguồn>):

- Sao chép chuỗi nguồn vào chuỗi đích.
- Chuỗi đích phải có đủ chiều dài!!
- strncpy(<chuỗi đích>, <chuỗi nguồn>, <độ dài sao chép>).

```
char s1[ ] = "Hello World";  
char s2[ 12 ];  
char s3[ ] = "Chao mung";
```

```
strcpy( s2, s1 );    // gán s1 vào s2.  
strcpy( s3, s1 );    // SAI, không đủ chiều dài.
```



■ Thư viện <string.h>:

■ `strcat(<chuỗi đích>, <chuỗi nguồn>)`:

- Nối chuỗi nguồn và cuối chuỗi đích.
- Chuỗi đích phải có đủ chiều dài!!

```
char s1[ ] = "Hello";  
char s2[ ] = "World";  
strcat( s1, s2 );      // nối s2 vào s1, SAI, không đủ chiều dài.
```

```
char s3[11];  
strcpy( s3, s1 );      // gán s1 vào s3.  
strcat( s3, s2 );      // nối s2 vào s3.
```



■ Thư viện <string.h>:

■ strcmp(s1, s2):

- So sánh s1 và s2 theo thứ tự từ điển.
- Trả về: 0 (bằng nhau), -1 (s1 < s2), +1 (s1 > s2).
- strcmp(s1, s2): không phân biệt hoa thường.

```
char s1[ ] = "abc";  
char s2[ ] = "abaab";  
char s3[ ] = "Abc";
```

```
int kq1 = strcmp(s1, s2); // kq1 = 1.  
int kq2 = strcmp(s3, s1); // kq2 = -1.  
int kq3 = strcmp(s3, s1); // kq2 = 0.
```

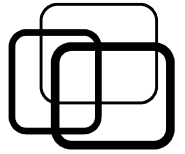


■ Thư viện <string.h>:

- `atoi(s)`: đổi chuỗi `s` ra số nguyên.
- `atof(s)`: đổi chuỗi `s` ra số thực.

```
char s1[ ] = "12345";  
char s2[ ] = "12.345";
```

```
int    x = atoi( s1);      // x = 12345.  
float  y = atof( s2);      // y = 12.345.
```



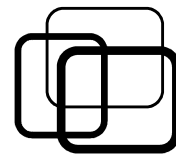
■ Chuỗi ký tự:

- Mảng ký tự kết thúc bằng '\0'.
- Nhập xuất: scanf, fgets, cin.getline, printf, puts.

■ Thư viện xử lý chuỗi:

- Thư viện <string.h>.
- strlen, strcmp, strcmp.
- strcpy, strcat.
- atoi, atof.

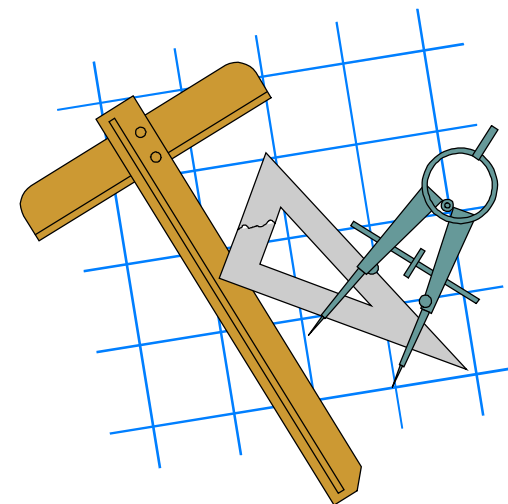


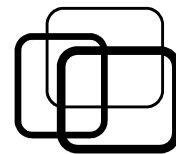


■ Bài tập 8.1:

Viết các hàm C giả lập các hàm trong thư viện `<string.h>`:

- `strcmp(s1, s2)`: so sánh thứ tự từ điển chuỗi `s1` và `s2`,
trả về: -1 ($s1 < s2$), $+1$ ($s1 > s2$), 0 ($s1 = s2$).
- `strcat(s1, s2)`: nối chuỗi `s2` vào cuối `s1`, `s1` phải đủ độ dài nối `s2`.
- `atof(s)`: đổi chuỗi `s` ra số thực.





■ Bài tập 8.2:

Viết chương trình C kiểm duyệt chuỗi:

(tổ chức theo dạng hàm và chia làm nhiều file):

- Nhập vào chuỗi ký tự S.
- Nhập vào chuỗi ký tự T cần kiểm duyệt.
- Xóa chuỗi T xuất hiện trong S và thay thế bằng chuỗi “###”.
- In kết quả.

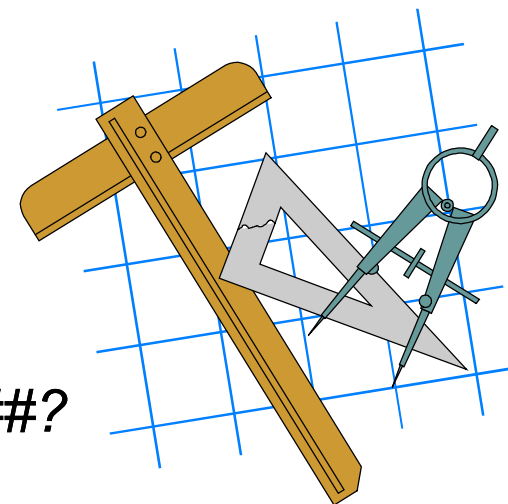
Định dạng nhập:

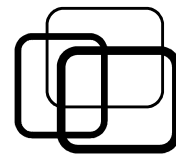
Nhap S = Nho ai, ai nho, bay gio nho ai?

Nhap T = ai

Định dạng xuất:

Kiem duyet = Nho ###, ### nho, bay gio nho ###?





■ Bài tập 8.3:

Viết chương trình C thống kê tần suất các ký tự trong chuỗi:
(tổ chức theo dạng hàm và chia làm nhiều file):

- Nhập vào chuỗi ký tự S.
- Đếm số lần xuất hiện các ký tự trong S.
- In ra số lần xuất hiện giảm dần cùng các ký tự.

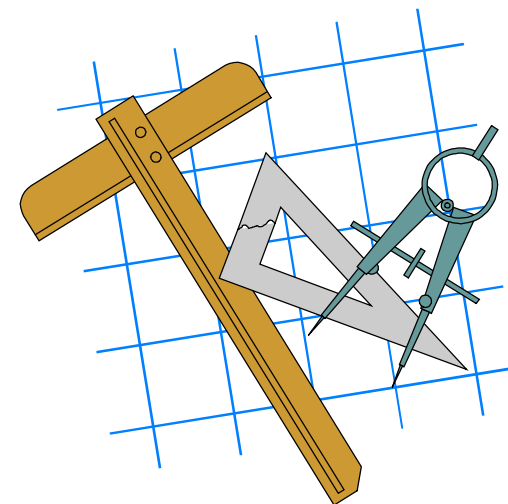
Định dạng nhập:

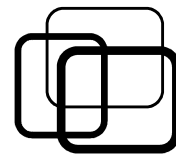
Nhap chuoi = tick tak tok

Định dạng xuất:

3: k t

1: a c i o





■ Bài tập 8.4:

Viết chương trình C xóa khoảng trắng dư thừa trong chuỗi:
(tổ chức theo dạng hàm và chia làm nhiều file):

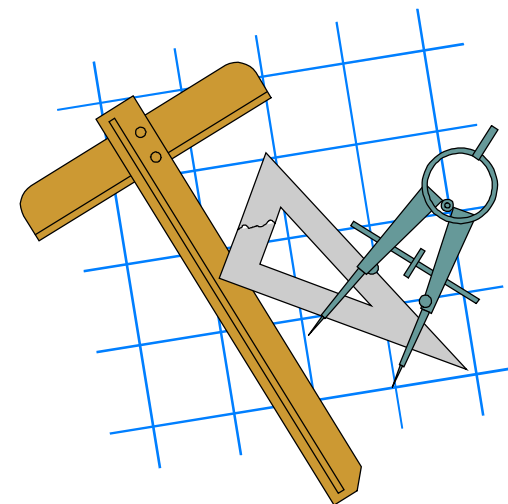
- Nhập vào chuỗi ký tự S.
- Bỏ khoảng trắng đầu chuỗi (trim left).
- Bỏ khoảng trắng cuối chuỗi (trim right).
- Bỏ khoảng trắng dư giữa các từ (giữ lại 1 khoảng trắng).

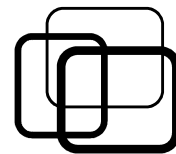
Định dạng nhập:

Nhap chuoi = hom nay troi dep qua

Định dạng xuất:

“hom nay troi dep qua”





■ Bài tập 8.5:

Viết chương trình C cắt chuỗi:

(tổ chức theo dạng hàm và chia làm nhiều file):

- Nhập vào chuỗi ký tự S.
- Nhập vào ký tự dùng để cắt C.
- Xuất danh sách các chuỗi con trong S đã được cắt bởi C.

Định dạng nhập:

Nhap chuoi = the,, quick, brown, fox,

Nhap ky tu = ,

Định dạng xuất:

“the”

“ quick”

“ brown”

“ fox”

