

# Chuỗi ký tự, kiểu cấu trúc

GV. Nguyễn Minh Huy

# Nội dung



- Chuỗi ký tự.
- Kiểu cấu trúc.

# Nội dung



- **Chuỗi ký tự.**
- **Kiểu cấu trúc.**



## ■ Chuỗi ký tự trong C:

- Chuỗi ký tự: 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 trên chuỗi ký tự:

### ■ Nhập chuỗi:

- `scanf("%s", &chuoi).`  
➔ Chỉ nhập từ đầu tiên.
- `gets(chuoi).`
- `fgets(chuoi, dodai, stdin);`  
➔ Nhập nguyên chuỗi.

### ■ Xuất chuỗi:

- `printf("%s", chuoi).`
- `puts(chuoi).`

```
#define MAX 100
```

```
int main()
```

```
{
```

```
    char s1[ MAX ];
```

```
    char s2[ MAX ];
```

```
    printf("Nhap chuoi s1 = ");
```

```
    scanf("%s", &s1);
```

```
    printf("Nhap chuoi s2 = ");
```

```
    gets(s2);
```

```
    printf("Chuoi s1 = %s", s1);
```

```
    puts(s2);
```

```
}
```



## ■ Thao tác trên chuỗi ký tự:

### ■ Thư viện <string.h>: **#include** <string.h>

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

```
char s1[ ] = "Hello World";           // Tự thêm '\0' ở cuối.  
char s2[ ] = "Hello World\n";         // Tự thêm '\0' ở cuối.  
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
```



## ■ Thao tác trên chuỗi ký tự:

### ■ Thư viện <string.h>: **#include** <string.h>

- strcpy( <chuỗi đích>, <chuỗi cần gán> ): gán chuỗi.  
➔ Chuỗi đích phải có đủ chiều dài để gán!!

```
char s1[ ] = "Hello World";           // Tự thêm '\0' ở cuối.  
char s2[ 12 ];  
char s3[ ] = "Chao mung";
```

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



## ■ Thao tác trên chuỗi ký tự:

### ■ Thư viện <string.h>:

- `strcat( s1, s2 )`: nối chuỗi `s2` vào cuối `s1`,  
s1 phải đủ độ dài nối `s2`.
- `strcmp( s1, s2 )`: so sánh thứ tự từ điển chuỗi `s1` và `s2`,  
trả về: -1 ( $s1 < s2$ ), +1 ( $s1 > s2$ ), 0 ( $s1 = s2$ ).
- `strcspn( s1, s2 )`: trả về vị trí đầu tiên xuất hiện `s2` trong `s1`,  
trả về độ dài `s1` nếu không tìm thấy.
- `atoi( s )`: đổi chuỗi `s` ra số nguyên.
- `atof( s )`: đổi chuỗi `s` ra số thực.



# Chuỗi ký tự



## ■ Thao tác trên chuỗi ký tự:

### ■ Viết hoa chữ cái đầu mỗi từ:

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

```
void vietHoa( char s[ ] )
{
    for (int i = 0; i < strlen(s); i++)
        if ( s[ i ] >= 'a' && s[ i ] <= 'z' &&
            ( i == 0 || s[ i - 1 ] == ' ' ) )
            s[ i ] = s[ i ] - 32;
}
```

```
#define MAX 100
```

```
int main()
{
    char s[MAX];

    printf("Nhap chuoi s = ");
    gets(s);

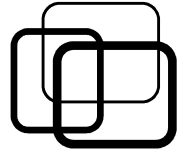
    vietHoa(s);

    printf("%s\n", s);
}
```

# Nội dung



- Chuỗi ký tự.
- **Kiểu cấu trúc.**



## ■ Xét chương trình sau:

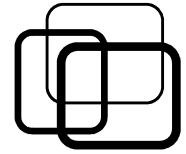
### ■ Thông tin một học sinh gồm:

- Họ tên.
- Ngày sinh.
- Giới tính.
- Điểm văn, toán, ngoại ngữ.

### ■ Viết chương trình:

- Nhập vào 1 học sinh.
- Xuất thông tin học sinh vừa nhập.

# Kiểu cấu trúc



## ■ Kiểu cấu trúc trong C:

### ■ Kiểu dữ liệu phức hợp.

➔ Gom nhóm dữ liệu với nhau.

### ■ Khai báo kiểu cấu trúc:

```
struct <Tên cấu trúc>
{
    <Khai báo thành phần 1>;
    <Khai báo thành phần 2>;
    ...
};
```

### ■ Khai báo biến cấu trúc:

```
<Tên cấu trúc> <Tên biến>;
```

```
struct HocSinh
{
    char hoten[50];
    char ngaysinh[11];
    bool gioitinh;
    float diemvan;
    float diemtoan;
};

int main()
{
    HocSinh hs1, hs2;
}
```

# Kiểu cấu trúc



## ■ Kiểu cấu trúc trong C:

### ■ Khởi tạo biến cấu trúc:

```
<Tên cấu trúc> <Tên biến> =  
{  
    <Giá trị thành phần 1>,  
    <Giá trị thành phần 2>,  
    ...  
};
```

### ■ Truy xuất thành phần:

```
<Tên biến> . <Tên thành phần>.
```

```
int main()  
{  
    HocSinh hs =  
    {  
        "Nguyen Van A",  
        "01/01/1997",  
        1,  
        7, 8, 9  
    };  
  
    hs.diemvan = 5;  
    hs.diemToan = 9;  
}
```



## ■ Kiểu cấu trúc trong C:

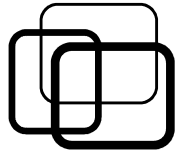
### ■ Truyền tham số:

- Giống các kiểu bình thường.
- Truyền tham trị
  - ➔ Giá trị thành phần không đổi.
- Truyền tham chiếu
  - ➔ Giá trị thành phần thay đổi.

```
void cong1(HocSinh hs)  
{  
    hs.diemvan++;  
    hs.diemtoan++;  
}
```

```
void cong2(HocSinh &hs)  
{  
    hs.diemvan++;  
    hs.diemtoan++;  
}
```

```
int main()  
{  
    HocSinh hs1;  
    cong1(hs1);  
    cong2(hs1);  
}
```



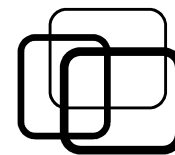
## ■ Chuỗi ký tự:

- Mảng ký tự kết thúc bằng '\0'.
- Nhập xuất: printf, scanf, gets, puts.
- Đếm chiều dài: strlen (thư viện <string.h>).
- Gán chuỗi: strcpy (thư viện <string.h>).

## ■ Kiểu cấu trúc:

- Kiểu dữ liệu phức hợp.
- Gom nhóm dữ liệu với nhau.
- Khai báo: từ khóa “**struct**”.
- Truy xuất phần tử: dấu “.”.

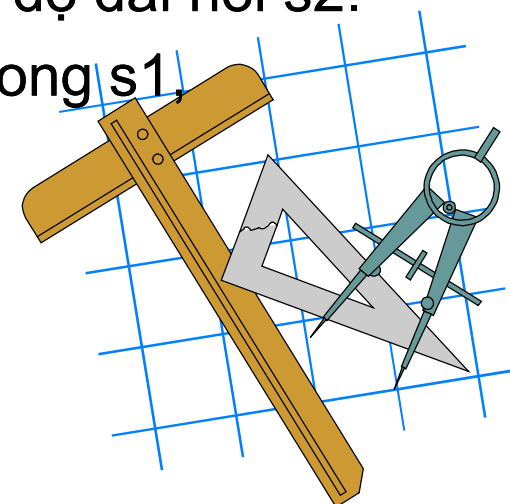




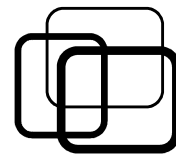
## ■ Bài tập 8.0:

Viết chương trình C giả lập các hàm trong thư viện `<string.h>`:  
(tổ chức theo dạng hàm và chia làm nhiều file):

- `strlen( s )`: trả về độ dài chuỗi `s`.
- `strcmp( s1, s2 )`: so sánh thứ tự từ điển chuỗi `s1` và `s2`,  
trả về: `-1` (`s1 < s2`), `+1` (`s1 > s2`), `0` (`s1 = s2`).
- `strcpy( s1, s2 )`: sao chép chuỗi `s2` vào `s1`, `s1` phải đủ độ dài chứa `s2`.
- `strcat( s1, s2 )`: nối chuỗi `s2` vào cuối `s1`, `s1` phải đủ độ dài nối `s2`.
- `strcspn( s1, s2 )`: trả về vị trí đầu tiên xuất hiện `s2` trong `s1`,  
trả về độ dài `s1` nếu không tìm thấy.
- `atoi( s )`: đổi chuỗi `s` ra số nguyên.
- `atof( s )`: đổi chuỗi `s` ra số thực.







## ■ Bài tập 8.1:

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.

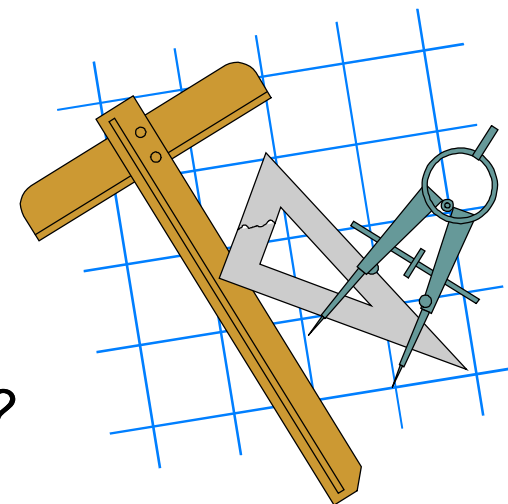
*Định dạng nhập:*

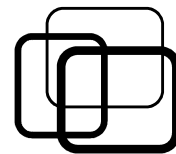
*Nhap S = Nho ai, ai nho, bay gio nho ai?*

*Nhap T = ai*

*Định dạng xuất:*

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





## ■ Bài tập 8.2:

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.

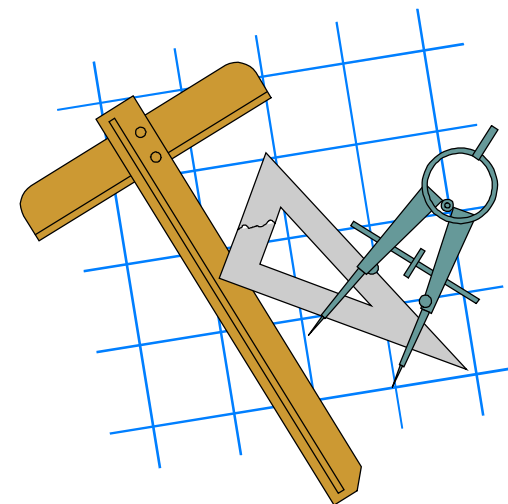
*Định dạng nhập:*

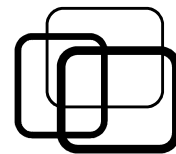
*Nhap chuoi = tick tak tok*

*Định dạng xuất:*

*3: k t*

*1: a c i o*





## ■ Bài tập 8.3:

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

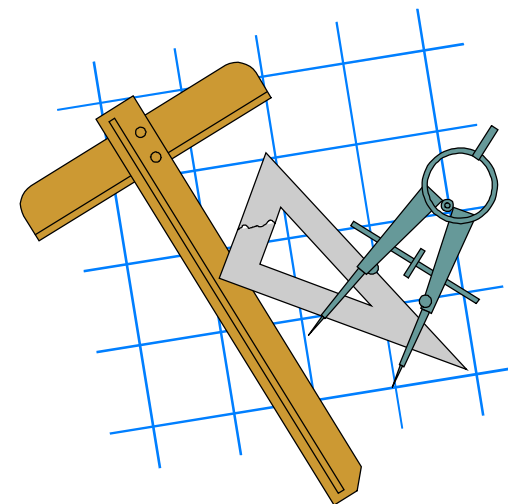
- Nhập vào chuỗi ký tự S.
- Bỏ khoảng trắng đầu chuỗi (trim left).
- Bỏ khoảng trắng cuối chuỗi (trim right).
- Bỏ khoảng trắng dư giữa các từ (giữ lại 1 khoảng trắng).

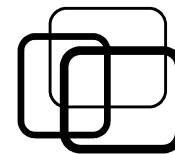
*Định dạng nhập:*

*Nhap chuoi =      hom    nay    troi    dep    qua*

*Định dạng xuất:*

*hom nay troi dep qua*

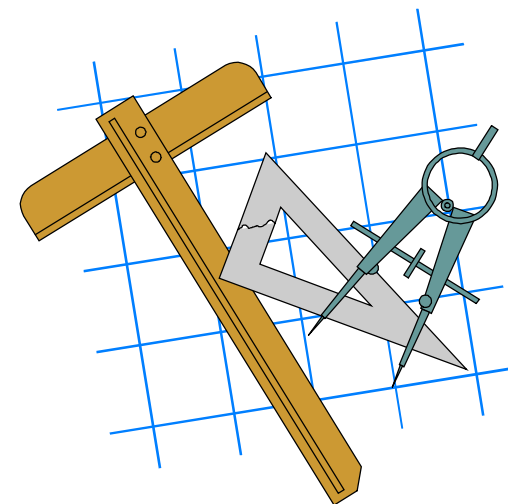




## ■ Bài tập 8.4:

Viết chương trình C thao tác phân số như sau:

- Khai báo kiểu cấu trúc phân số.
- Nhập vào 2 phân số.
- Tính và xuất kết quả tổng, tích, nghịch đảo, rút gọn của 2 phân số vừa nhập.





## ■ Bài tập 8.5:

Viết chương trình C thao tác học sinh như sau:

- Khai báo kiểu cấu trúc học sinh (như bài học).
- Nhập vào danh sách N học sinh.
- Xuất danh sách học sinh giỏi (điểm trung bình  $\geq 8$ ) theo thứ tự điểm trung bình giảm dần.

