



CƠ SỞ LẬP TRÌNH

Kiểu chuỗi ký tự

- ☐ Khai báo
- ☐ Các thao tác trên chuỗi ký tự
- ☐ Mảng chuỗi ký tự
- ☐ Một số hàm xử lý chuỗi ký tự

1. Khai báo

□ Khái niệm

- Chuỗi ký tự trong C được xây dựng như mảng một chiều các ký tự
- Chuỗi ký tự kết thúc bằng ký tự `'\0'` (ký tự NULL trong bảng mã ASCII)
 - ➔ Độ dài tối đa của chuỗi = kích thước mảng – 1
 - ➔ Khai báo nên dành ra 1 ô nhớ để chứa ký tự `'\0'`

□ Ví dụ

```
char line[80];    // Dài tối đa 79 ký tự
char hoten[30];   // Dài tối đa 29 ký tự
```

Khai báo có khởi tạo giá trị

□ Độ dài cụ thể

```
char string[40]="Ngon ngu C";
```

- Khởi tạo Chuỗi ký tự có độ dài tối đa 39 ký tự với giá trị ban đầu là chuỗi “Ngon ngu C”

□ Tự xác định độ dài

```
char str[]="Ngon ngu C";
```

- Chương trình dịch tự bố trí một mảng để chứa dãy ký tự và 1 ô chứa kí hiệu ‘\0’

□ Chú ý:

- Khai báo Chuỗi ký tự với con trỏ

```
char *message;
```

```
message="Xin chao!";
```

2. Các thao tác trên chuỗi ký tự

- ☐ Nhập chuỗi từ bàn phím
- ☐ Xuất chuỗi ra màn hình
- ☐ Xác định độ dài chuỗi
- ☐ Ghép chuỗi
- ☐ Sao chép chuỗi
- ☐ So sánh chuỗi
- ☐ Tìm kiếm ký tự
- ☐ ...

a) Nhập chuỗi từ bàn phím

- ❑ Sử dụng hàm **scanf** với đặc tả “%s”

scanf (“%s”, **str**) ;

- Chỉ nhận các ký tự từ bàn phím đến khi gặp ký tự dấu cách, tab, ký tự xuống dòng.
- chuỗi nhận được không bao gồm dấu cách

- ❑ Ví dụ:

```
char monhoc[50];  
printf("Nhap mot xau ki tu: ");  
scanf("%s", monhoc);  
printf("Xau nhan duoc la: %s", monhoc);
```

```
Nhap mot chuoì: Ngon ngu lap trinh C  
Chuoì nhan duoc la: Ngon_
```

a) Nhập chuỗi từ bàn phím (tt)

□ Sử dụng hàm `gets`

`gets(str) ;`

- Nhận các ký tự từ bàn phím đến khi gặp ký tự xuống dòng.
- chuỗi nhận được là những gì người dùng nhập (trừ ký tự xuống dòng).

□ Ví dụ

```
char monhoc[50] ;  
printf("Nhap mot chuoi: ");  
gets(monhoc) ;  
printf("Chuoi nhan duoc la: %s", monhoc) ;
```

```
Nhap mot chuoi: Ngon ngu lap trinh C  
Chuoi nhan duoc la: Ngon ngu lap trinh C_
```

b) Xuất chuỗi ra màn hình

- Sử dụng hàm `printf` với đặc tả “%s”

```
char monhoc[50] = "Ngon ngu C";  
printf("%s", monhoc);
```

```
Ngon ngu C_
```

- Sử dụng hàm `puts`

```
char monhoc[50] = "Ngon ngu C";  
puts(monhoc);
```

```
Ngon ngu C
```

```
_
```

⇔ `printf("%s\n", monhoc);`

c) Xác định độ dài chuỗi

□ Tự xác định

Đếm cho đến khi gặp ký tự '\0'

```
char str[]="Ngon ngu C";  
int dem=0;  
while (str[dem]!='\0') dem++;  
printf("Do dai chuoi la: %d ky tu", dem);
```

□ Sử dụng con trỏ để xử lý chuỗi

```
char *message;  
message="Ngon ngu C";  
int dem=0;  
while (*message!='\0') { *message++; dem++; }  
printf("Do dai chuoi la: %d ky tu", dem);
```

c) Xác định độ dài chuỗi (tt)

□ Dùng hàm

Hàm `strlen(str)` trong thư viện `string.h`

```
printf("Do dai chuoi  
la: %d", strlen(str));
```

□ Bài tập:

1. Nhập từ bàn phím chuỗi `st1`, viết ra màn hình chuỗi đó theo chiều ngược lại.
2. Nhập từ bàn phím chuỗi `st2`, chuyển chuỗi `st2` sang chữ hoa và viết ra màn hình chuỗi kết quả ra màn hình.

d) Ghép chuỗi

□ Ghép chuỗi st2 vào sau chuỗi st1

- Hàm `strcat(st1,st2)`: nối chuỗi st2 vào sau chuỗi st1

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

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

```
#define maxst 40
```

```
main()
```

```
{   char st1[maxst]="Chao mung";
```

```
    char st2[maxst]="Ngon ngu C";
```

```
    if (maxst>strlen(st1)+strlen(st2))
```

```
        puts(strcat(st1,st2));
```

```
    else printf("Khong du bo nho!");
```

```
}
```

- Chú ý: không viết `st=st1+st2;`

e) Sao chép chuỗi

- ❑ C không cho phép gán một chuỗi cho một biến do Chuỗi ký tự là một mảng.
 - Ví dụ: Không viết được `line="Hello";`
 - ➔ Dùng hàm sao chép chuỗi hoặc viết vòng lặp sao chép từng ký tự
- ❑ Hàm sao chép chuỗi
 - **strcpy**(st1, st2) – gán chuỗi st2 cho chuỗi st1
 - Ví dụ: `strcpy(line, "Hello");`
- ❑ Chú ý:
 - Hàm không kiểm tra tính đúng đắn về kích thước ô nhớ của st1 có đủ chứa st2 hay không, do đó, cần lưu ý về kích thước Chuỗi ký tự.

f) So sánh chuỗi ký tự

- ❑ Quy tắc so sánh
 - Các ký tự của 2 chuỗi được so sánh từng cặp từ trái qua phải theo giá trị của mã ASCII
- ❑ Hàm `strcmp(st1, st2)` trong thư viện `string.h`
 - Trả về 0 nếu `st1==st2`
 - `<0` nếu `st1<st2`
 - `>0` nếu `st>st2`
- ❑ Ví dụ: Nhập các chuỗi cho đến khi nhập chuỗi “done”

```
char st[80];  
do  
{  
    gets(st);  
    printf("Chuoi vua nhap: %s \n",st);  
} while (strcmp(st, "done"));
```

f) So sánh chuỗi ký tự (tt)

□ Một số hàm so sánh khác

- **strcmp(st1, st2)** ; so sánh chuỗi st1 với st2 nhưng không phân biệt chữ hoa-thường.
- **strncmp(st1, st2, n)** ; so sánh n ký tự đầu tiên của st1 và st2.
- **strnicmp(st1, st2, n)** ; so sánh n ký tự đầu tiên của st1 và st2 nhưng không phân biệt chữ hoa-thường.

g) Tìm kiếm ký tự

❑ Hàm `strchr(str, c)` tìm kiếm ký tự `c` trong chuỗi `str`

- Kết quả là con trỏ trỏ tới vị trí của ký tự `c`

- Nếu không tìm thấy trả về NULL

```
char str[80];    char c;  
printf("Nhap chuoi: "); gets(str);  
printf("Nhap ky tu can tim: "); c=getchar();  
if (strchr(str, c)) printf("Tim thay %c");  
else printf("Khong tim thay!");
```

❑ Hàm `strstr(str1, str2)` tìm kiếm chuỗi `str2` trong chuỗi `str1`

- Kết quả là con trỏ trỏ tới vị trí của chuỗi `str2`

- Nếu không tìm thấy trả về NULL

3. Mảng chuỗi ký tự

□ Bài toán:

- Nhập mảng các chuỗi ký tự, sắp xếp các chuỗi đó theo thứ tự từ điển
- Sử dụng mảng 2 chiều để lưu các chuỗi ký tự

```
void sapxep(int n, char x[][80])
{
    char temp[80];
    int i, j;
    for (i=0; i<n-1; i++)
        for (j=i+1; j<n; j++)
            if (strcmp(x[i], x[j])>0)
            {
                strcpy(temp, x[i]);
                strcpy(x[i], x[j]);
                strcpy(x[j], temp);
            }
}
```


Mảng Chuỗi ký tự (tt)

```
int main()
{
    char st[40][80];
    int i,n=0;
    printf("Nhap cac chuoi ki tu, ket thuc bang chu
    \n\"end\"\\n");
    do
    {
        printf("Nhap chuoi thu %d: ",n+1);
        gets(st[n]);
    }
    while (strcmp(st[n++], "end"));
    n--;
    sapxep(n,st);
    printf("\\Day cac chuoi ki tu sau khi sap xep\\n");
    for(i=0;i<n;i++) puts(st[i]);
}
```

4. Một số hàm xử lý chuỗi ký tự

□ Các hàm trong thư viện `string.h`

- Hàm `strlwr(st)`: chuyển chuỗi `st` thành chữ thường
- Hàm `strupr(st)`: chuyển chuỗi `st` thành chữ hoa
- Hàm `strrev(st)`: đảo ngược chuỗi `st`

□ Các hàm xử lý ký tự trong `ctype.h`

- Hàm `toupper(c)`: chuyển `c` thành chữ hoa
- Hàm `tolower(c)`: chuyển `c` thành chữ thường
- Hàm `isalpha(c)`: đúng (khác 0) nếu `c` là chữ cái
- Hàm `islower(c)`: đúng nếu `c` là chữ cái thường
- Hàm `isupper(c)`: đúng nếu `c` là chữ cái hoa
- Hàm `isspace(c)`: đúng nếu `c` là dấu cách, dấu `\n`, dấu về đầu dòng `\r`, tab `\t`

Một số hàm xử lý chuỗi ký tự (tt)

- ❑ Các hàm chuyển đổi (trong `stdlib.h`)
 - Hàm `atoi(str)`: chuyển đổi chuỗi `str` thành số nguyên `int`
 - ❑ Ví dụ: Đọc một số nguyên có thể dùng cặp lệnh
`gets(str) ;`
`n=atoi(str) ;` //tránh đọc xong số mà bộ đệm vẫn còn `\n`
 - Hàm `atol(str)`: chuyển đổi chuỗi `str` thành số nguyên `long`
 - Hàm `atof(str)`: chuyển đổi `str` thành số thực `float`
- Các hàm này bỏ qua các dấu cách ở đầu, chuyển cho đến khi gặp ký tự không thích hợp, nếu không chuyển được thì kết quả là 0

Bài tập thực hành

- ❑ Bài 1. Viết hàm `upper(char s[])` đổi toàn bộ các ký tự sang ký tự hoa (giống hàm `strupr`)
- ❑ Bài 2. Viết hàm `lower(char s[])` đổi toàn bộ các ký tự sang ký tự thường (giống hàm `strlwr`)
- ❑ Bài 3. Viết hàm `proper(char s[])` đổi các ký tự đầu tiên của mỗi từ sang ký tự hoa.
- ❑ Bài 4. Đếm xem có bao nhiêu từ trong `s` (‘từ’ là tập hợp các ký tự in được không chứa các dấu cách, xuống dòng, tab). In ra màn hình các từ trong `s`.
- ❑ Bài 5. Viết các hàm `left`, `right`, `mid...`

Bài tập thực hành

- ❑ Bài 6. Tìm từ có độ dài lớn nhất trong chuỗi s.
- ❑ Bài 7. Xóa tất cả các khoảng trắng của s
- ❑ Bài 8. Viết hàm `standard(char s[])` loại bỏ toàn bộ khoảng trắng đầu chuỗi, cuối chuỗi và giữa 2 từ trong s chỉ còn 1 khoảng trắng.
- ❑ Bài 9. Đếm số lượng từ 'em' trong Chuỗi ký tự được đọc từ bàn phím.
- ❑ Bài 10. Kiểm tra xem từ có phải là palindrome hay không? Từ là palindrome là từ đọc từ phải sang trái cũng giống như đọc từ trái sang phải.
Ví dụ: civic, madam, level,...