



TRƯỜNG ĐẠI HỌC THỦY LỢI  
Khoa Công nghệ thông tin  
Bộ môn Tin học và KTTT

# **NHẬP MÔN LẬP TRÌNH**

Giảng viên: TS.GVC Nguyễn Quỳnh Diệp

Email: [diepnq@tlu.edu.vn](mailto:diepnq@tlu.edu.vn)

Điện thoại: 0904345673

# Chương 4: Mảng, con trỏ và chuỗi ký tự

## 4.1. Mảng

- Khái niệm
- Khai báo và sử dụng
- Các thao tác thường gặp

## 4.2. Con trỏ

- Khái niệm và cách khai báo
- Toán tử địa chỉ (&), toán tử nội dung (\*)
- Phép toán trên con trỏ
- Con trỏ và mảng

## 4.3. Chuỗi ký tự

- Khái niệm, khai báo và sử dụng
- Các hàm xử lý ký tự và chuỗi ký tự
- Mảng chuỗi ký tự



## 4.3. XÂU

---

- **Xâu kí tự** (string) là một dãy các **kí tự** viết liên tiếp nhau
  - Độ dài xâu là số kí tự có trong xâu
  - Xâu không có kí tự nào gọi là **xâu rỗng**
- Ví dụ: "Tin hoc", "String"
- Lưu trữ: kết thúc xâu bằng kí tự **'\0'** hay NULL (mã ASCII là 0)

'T'	'i'	'n'	' '	'h'	'o'	'c'	'\0'
-----	-----	-----	-----	-----	-----	-----	------

- **Xâu kí tự >< mảng kí tự**
  - Xâu kí tự là tập hợp các kí tự viết liên tiếp nhau
    - Truy nhập một phần tử của xâu ký tự (*là một ký tự*) giống như truy nhập vào một phần tử của mảng: Tên[Chỉ\_số]
  - Xâu kí tự có kí tự kết thúc xâu, mảng kí tự không có kí tự kết thúc xâu
- **Xâu kí tự độ dài 1 >< kí tự ("A" == 'A' ?)**
  - 'A' là 1 kí tự, được lưu trữ trong 1 byte
  - "A" là 1 xâu kí tự, ngoài kí tự 'A' còn có kí tự '\0' => được lưu trữ trong 2 byte

`char tên_xâu [số_kí_tự_tối_đa];`

- Để lưu trữ một xâu có  $n$  kí tự chúng ta cần một mảng có kích thước  $n+1$ 
  - Phần tử cuối cùng chứa ký tự `'\0'` hoặc NULL

## Ví dụ

- Để lưu trữ xâu "Tin hoc" chúng ta phải khai báo xâu có số phần tử tối đa ít nhất là 8

```
char str[8] = "Tin hoc";
```

- Giống như truy cập tới một phần tử của mảng kí tự:

tên\_xâu [chỉ\_số\_của\_kí\_tự]

- **Ví dụ:** char Str[10] = "Tin hoc";

T	i	n		h	o	c	\0	?	?
---	---	---	--	---	---	---	----	---	---

Str[0] → 'T'

Str[8] → ?

Str[3] → ' '

Str[9] → ?

Str[7] → '\0'

- Nhập 1 xâu và đếm số kí tự 'a' trong xâu

```
#include <stdio.h>
int main(){
    char str[100];
    int d = 0, i = 0;
    printf("Nhap xau ky tu: "); gets(str);
    while(str[i]!='\0'){
        if(str[i]=='a')
            d++;
        i++;
    }
    printf("Ket qua: %d",d);
    return 0;
}
```

```
Nhap xau ky tu: Truong Dai hoc Bach khoa Ha Noi 2021!
Ket qua: 4
-----
Process exited after 18.9 seconds with return value 0
Press any key to continue . . .
```



- Nhập 1 câu và in ra các từ, mỗi từ trên 1 dòng

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

```
int main(){
```

```
    char S[100];
```

```
    int i = 0;
```

```
    printf("Nhap xau: "); gets(S);
```

```
    while(S[i]!='\0')
```

```
    {
```

```
        //32 là mã ASCII của phím Space
```

```
        if(S[i] != 32 && S[i+1]==32)
```

```
            printf("%c\n", S[i]);
```

```
        else
```

```
            if(S[i] != 32) printf("%c",S[i]);
```

```
        i++;
```

```
    }
```

```
    return 0;
```

```
}
```

```
Nhap xau: Chao mung cac ban den voi DHBK
Chao
mung
cac
ban
den
voi
DHBK
-----
```

Đếm số từ, nếu các từ được cách nhau bởi dấu phân cách

- Tập tiêu đề: `ctype.h`

`#include <ctype.h>`

- `int toupper(char ch)`: Chuyển kí tự thường thành kí tự hoa

`toupper('a') => 'A'`

- `int tolower(char ch)`: Chuyển kí tự hoa thành kí tự thường

`tolower('B') => 'b'`

- **int islower(char ch):** Kiểm tra chữ thường
  - Hàm trả về giá trị khác 0 nếu ch là chữ thường, ngược lại trả về 0
  - Ví dụ: `printf("%d ", islower('A'));`  $\Rightarrow 0$
- **int isupper(char ch):** Kiểm tra chữ hoa
  - Hàm trả về giá trị khác 0 nếu ch là chữ hoa, ngược lại trả về 0
  - Ví dụ: `printf("%d ", isupper('A'));`  $\Rightarrow \neq 0$

- **int isalpha(char ch):** Kiểm tra kí tự có phải chữ cái không ('a'...'z','A'..'Z').
  - Hàm trả về khác 0 nếu đúng, ngược lại trả về giá trị bằng 0
  - Ví dụ: `printf("%d ",isalpha('A'));`  $\Rightarrow \neq 0$
- **int isdigit(char ch):** Kiểm tra kí tự có phải chữ số ('0','1',...'9') không.
  - Hàm trả về khác 0 nếu đúng, ngược lại trả về giá trị bằng 0
  - Ví dụ: `printf("%d ",isdigit('A'));`  $\Rightarrow 0$



# Các hàm xử lý kí tự



- **int isspace(char ch):** Kiểm tra kí tự dấu cách (mã 32), xuống dòng ('\n' 10), đầu dòng ('\r' 13), tab ngang ('\t' 9), tab dọc ('\v' 11).
  - Hàm trả về khác 0 nếu đúng, ngược lại trả về giá trị bằng 0
- **int iscntrl(char ch):** Kiểm tra kí tự điều khiển (0-31).
  - Hàm trả về khác 0 nếu đúng, ngược lại trả về giá trị bằng 0

- Nhập vào 1 xâu và chuyển thành xâu chữ hoa

```
#include <stdio.h>
#include <ctype.h>
int main(){
```

```
    char S[50];
    int i = 0;
    printf("Nhap mot xau: "); gets(S);
    printf("\nXau ban dau: %s.",S);
    while(S[i]!= '\0'){
        S[i] = toupper(S[i]);
        i = i + 1;
    }
    printf("\nXau ket qua: %s",S);
    return 0;
}
```

```
Nhap mot xau: hoc nua, hoc nua, hoc mai!
```

```
Xau ban dau: hoc nua, hoc nua, hoc mai!.
```

```
Xau ket qua: HOC NUA, HOC NUA, HOC MAI!
```

```
-----
```

```
Process exited after 23.1 seconds with return value 0
Press any key to continue . . .
```

## ■ Nhập 1 xâu và đếm các từ trong xâu

```
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
int main(){
    char s[100];
    int d = 0, i = 0;
    printf("Nhap xau ky tu: "); gets(s);
    if(s[0] == '\\0')
        printf("Xau rong!");
    else{
        if(!isspace(s[0]))
            d = 1;
        i = 1;
        while(s[i] != '\\0'){
            if(isspace(s[i-1]) && (!isspace(s[i])))
                d++;
            i++;
        }
        printf("Ket qua: %d",d);
    }
    return 0;
}
```

```
Nhap xau ky tu: Dai hoc Bach Khoa HN
Ket qua: 5
```

```
-----
Process exited after 11.33 seconds with return value 0
Press any key to continue . . .
```

## ▪ Cách 2

```
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
int main(){
    char s[100];
    int d = 0, i = 0;
    printf("Nhap xau ky tu: "); gets(s);
    for (i = 0; s[i] != '\0'; i++)
        if (s[i] != ' ' && s[i+1] == ' ' ||
            s[i] != '\0' && s[i+1] == '\0')
            d++;
    if (d==0)
        printf("Xau rong!");
    else
        printf("Ket qua: %d",d);
    return 0;
}
```



## ❖ Vào/ra chuỗi ký tự

- Tập tiêu đề: `stdio.h`

`#include <stdio.h>`

- Nhập chuỗi ký tự
  - `gets(tên_xâu);`
  - `scanf("%s",&tên_xâu);`
- Hiển thị chuỗi ký tự
  - `puts(tên_xâu);`
  - `printf("%s",tên_xâu);`

Sự khác nhau giữa `gets` và `scanf`?

- Hàm `gets` không kiểm tra kích thước chuỗi được nhập

```
#include <stdio.h>
#include <ctype.h>
int main(){
    char s[10];
    int i = 0;
    printf("Nhap xau ky tu: "); gets(s);
    printf("Xau da nhap: %s",s);
    return 0;
}
```

Khai báo chuỗi ký tự có tối đa 10 ký tự, nhưng có thể nhập nhiều hơn 10 ký tự với `gets()`

```
Nhap xau ky tu: Chao mung cac sinh vien den voi DHBK
Xau da nhap: Chao mung cac sinh vien den voi DHBK
-----
```

- Giải pháp: dùng `fgets(str, so_ky_tu_toida, stdin);`

- Giải pháp: dùng `fgets(str, so_ky_tu_toida, stdin);`
- Ví dụ:

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

```
int main(){
```

```
    char s[10], w[10];
```

```
    printf("Nhap xau ky tu s: "); gets(s);
```

```
    printf("Nhap xau ky tu w: "); fgets(w,10,stdin);
```

```
    printf("\nXau da nhap s: %s", s);
```

```
    printf("\nXau da nhap w: %s", w);
```

```
    return 0;
```

```
}
```

```
Nhap xau ky tu s: Dai hoc Bach Khoa Hn
Nhap xau ky tu w: Dai hoc Bach Khoa HN

Xau da nhap s: Dai hoc Bach Khoa Hn
Xau da nhap w: Dai hoc B
-----
```

- Tập tiêu đề: string.h

`#include <string.h >`

## Chú ý:

```
char str[100] = "Hello world";
```

```
char * p = str;
```

- p là con trỏ, trỏ tới mảng các ký tự/xâu ký tự
- p+6: (Phép tính toán trên con trỏ), p+6 trỏ tới xâu "world".
- Xâu ký tự, có thể được khai báo `char *`

- `size_t strlen(const char * chuỗi)`
  - Trả về độ dài chuỗi

```
printf("%d ",strlen("Hello world")); //11
```
- `char * strcpy(char * đích, const char * nguồn)`
  - Sao chép nội dung chuỗi *nguồn* vào chuỗi *đích*, trả về giá trị chuỗi nguồn

```
char Str[20];  
printf("%s",strcpy(Str,"Hello")); //Hello  
printf("%s", Str); //Hello
```

Chú ý: Phép gán `Str = "Hello"` là không hợp lệ

- `int strcmp(const char * xâu_1, const char * xâu_2)`
  - So sánh hai chuỗi.
  - Trả về giá trị 0 nếu hai chuỗi giống nhau;
  - Giá trị  $< 0$ : `xâu_1 < xâu_2`
  - Giá trị  $> 0$ : `xâu_1 > xâu_2`
- Ví dụ
  - `char Str[20];`
  - `strcpy(Str, "hello");` → *Str = "hello"*
  - `printf("%d", strcmp(Str, "hello"));` → 0
  - `printf("%d", strcmp(Str, "hello!"));` → -1 (!?)
  - `printf("%d", strcmp(Str, "Hello"));` → 1 (!?)

*hello, Hello*  
*↑            ↑*  
*104        72*  
*>*

- `char *strcat(char *đích, const char *nguồn)`
  - Ghép nối chuỗi nguồn vào ngay sau chuỗi đích, trả lại chuỗi kết quả

## Ví dụ

```
char Str[20];
```

```
strcpy(Str, "Hello ");
```

```
printf("%s ", strcat(Str, "world")); ⇒ Hello world
```

```
printf("\n%s", Str); ⇒ Hello world
```

- `char * strchr (const char * s, int c)` *char*
  - Trả về con trỏ trỏ tới vị trí xuất hiện đầu tiên của ký tự c trong s. Nếu không có trả về con trỏ null

```
strcpy(Str, "Hello world");  
printf("%s ", strchr(Str, 'o')); ⇒ o world
```
- `char* strstr (const char * s1, const char * s2)`
  - Trả về con trỏ trỏ tới vị trí xuất hiện đầu tiên của chuỗi s2 trong s1. Nếu không tồn tại, trả về con trỏ null

```
printf("%s ", strstr(Str, "llo")); ⇒ llo world
```





# Lưu ý:



- Tương tự như hàm `gets`, một số hàm xử lý xâu: `strcpy`, `strcat` cũng đã không dùng được đối với các trình biên dịch mới.
- Tuy nhiên trong phạm vi môn Nhập môn lập trình, chúng ta vẫn chấp nhận dùng các hàm trên với trình biên dịch phù hợp.

- Nhập 2 xâu, cho biết số lần xuất hiện xâu 2 trong xâu 1.

```
Nhap xau thu nhat: abcabcabcdefgh
Nhap xau thu hai: abc
Xau "abc" hien trong xau "abcabcabcdefgh" 3 lan
-----
```

```
Nhap xau thu nhat: hoc nua hoc mai hoc suot doi
Nhap xau thu hai: hoc
Xau "hoc" hien trong xau "hoc nua hoc mai hoc suot doi" 3 lan
-----
```

```
#include <stdio.h>
#include <string.h>
int main(){
    int d = 0;
    char S1[100], S2[50], *p;
    printf("Nhap xau thu nhât: "); gets(S1);
    printf("Nhap xau thu hai: "); gets(S2);
    p = strstr(S1,S2);
    while(p != NULL){
        d = d + 1;
        p = strstr(p+1,S2); //vi tri tim kiem
ke tiep
    }
    printf("Xau \"%s\" hien trong xau \"%s\" %d
lan",S2,S1,d);
    return 0;
}
```

*S<sub>1</sub> = abcabcabcdegg    S<sub>2</sub> = abc*  
*↑    ↑    ↑*  
*p    p*

*S<sub>2</sub> = gt*

```
Nhap xau thu nhât: hoc nua hoc mai hoc suot doi
Nhap xau thu hai: hoc
Xau "hoc" hien trong xau "hoc nua hoc mai hoc suot doi" 3 lan
-----
```

- Kiểm tra chuỗi đối xứng hay không

```
Nhap vao xau ki tu: abccba
Xau doi xung!
```

```
Nhap vao xau ki tu: abcdegfh
Xau khong doi xung!
```

```
#include<stdio.h>
#include<string.h>
int main(){
    char s[50];
    printf("Nhap vao xau ki tu: "); gets(s);
    int i, n = strlen(s);
    for(i = 0; i < n/2; i++)
        if(s[i] != s[n-1-i])
            break;
    if(i == n/2)
        printf("Xau doi xung!");
    else
        printf("Xau khong doi xung!");
    return 0;
}
```

```
Nhap vao xau ki tu: abcba
Xau doi xung!
-----
```

```
Nhap vao xau ki tu: abcdegfh
Xau khong doi xung!
-----
```

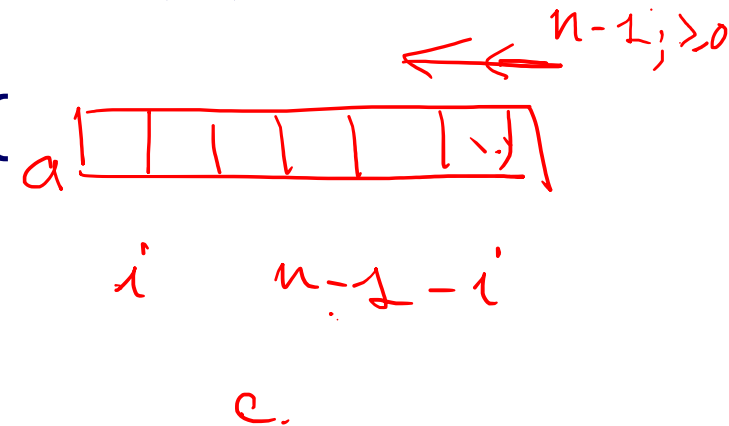
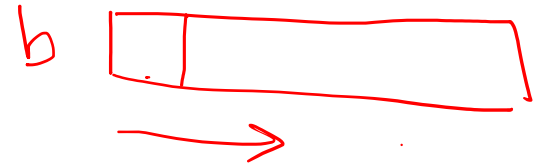
Handwritten diagram illustrating string reversal logic for the example "abcba":

The string "abcba" is shown with indices  $i=0$  and  $i=1$  on the left, and  $n-1-1$  and  $n-1-0$  on the right. Arrows indicate the swapping of characters at these indices.

- Nhập 1 chuỗi và đảo ngược chuỗi đó

```
Nhap xau: Dai hoc Bach Khoa HN
Xau dao nguoc: NH aohK hcaB coh iaD
-----
Process exited after 6.468 seconds with return value 0
Press any key to continue . . .
```

```
#include<stdio.h>
#include<string.h>
int main(){
    char s[100], c;
    printf("Nhap xau: "); gets(s);
    int i, n = strlen(s);
    for(i = 0; i<n/2; i++){
        c = s[i];
        s[i] = s[n-i-1];
        s[n-i-1] = c;
    }
    printf("Xau dao nguoc: %s",s);
    return 0;
}
```





- **Xâu ký tự có thể là kiểu của phần tử trong mảng**
- **Khai báo**
  - `char DS[100][30];`
  - Mảng có tối đa 100 phần tử, các phần tử là xâu có độ dài tối đa 30
- **Sử dụng**
  - Như một mảng bình thường
  - Mỗi phần tử mảng được sử dụng như một xâu ký tự

Thank you!

DS

	Thursday	10/1
	- 1 1 4	1

Diagram illustrating a memory layout or data structure. It shows a horizontal bar divided into two sections. The left section is labeled "Ds" and contains four vertical lines. The right section is labeled "95". Above the bar, there are two "0" labels, one above the left section and one above the right section.



- Nhập vào DSSV tối đa 10 phần tử cho tới khi gặp tên rỗng, in Danh sách vừa nhập.

```
Nhap DSSV (<100), go Enter de thoat..  
Ten sinh vien [1]: Hoang Mai  
Ten sinh vien [2]: Tran Anh  
Ten sinh vien [3]: Nguyen Hoang  
Ten sinh vien [4]: Mai Hong  
Ten sinh vien [5]:  
  
DS sinh vien vua nhap:  
Hoang Mai  
Tran Anh  
Nguyen Hoang  
Mai Hong  
-----
```



## ▪ Cách 2:

```
#include <stdio.h>
#include <string.h>
int main(){
    int i, j;
    char DS[10][30];
    printf("Nhap DSSV (<10), go Enter de thoat..\n");
    for(i = 0; i <10; i++){
        printf("Ten sinh vien [%d]: ", i+1); gets(DS[i]);
        if(DS[i][0] == '\0') break;
    }
    printf("\nDS sinh vien vua nhap:");
    for(j=0; j<i; j++)
        printf("\n%s", DS[j]);
    return 0;
}
```

- Nhập dãy (<100) xâu cho tới khi gặp xâu "\*\*\*".  
Đưa ra màn hình xâu có độ dài lớn nhất

```
Nhap xau thu [1]: Ha Noi
Nhap xau thu [2]: Thai Nguyen
Nhap xau thu [3]: Thai Binh
Nhap xau thu [4]: Ha Giang
Nhap xau thu [5]: Hoa Binh
Nhap xau thu [6]: ***

Xau dai nhat la: Thai Nguyen, co do dai: 11.
-----
```

```
#include <stdio.h>
#include <string.h>
int main(){
    int i = 0;
    char DS[10][30], s[30] = "";
    do{
        printf("Nhap xau thu [%d]: ",i+1); gets(DS[i]);
        if(strcmp(DS[i],"***")!=0) //xau DS[i] <> "***"
            i = i + 1;
        else
            break;
    }while(i<10);
    int j, d = 0;
    for(j = 0; j < i; j++){
        if(strlen(DS[j]) > d){
            d = strlen(DS[j]);
            strcpy(s,DS[j]);
        }
    }
    printf("\nXau dai nhat la: %s, co do dai: %d.",s,d);
    return 0;
}
```

- Nhập vào DS họ tên sinh viên, in ra DS sau khi đã sắp xếp tăng dần

```
Số sinh viên: 3
Tên sinh viên [1]: Tran Mai
Tên sinh viên [2]: Hoang Hoa
Tên sinh viên [3]: Nguyen Ngan

DS sinh viên vừa nhập:
Hoang Hoa
Nguyen Ngan
Tran Mai
```

-----

# Ví dụ 11: code

```
#include <stdio.h>
#include <string.h>
int main(){
    int i, j, N;
    char DS[20][30], s[30];
    //Nhap DS sinh vien
    printf("So sinh vien: "); scanf("%d",&N);
    fflush(stdin);
    for(i = 0; i<N; i++){
        printf("Ten sinh vien [%d]: ",i+1); gets(DS[i]);
    }
    //Sap xep tang dan
    for(i = 0; i<N-1; i++)
        for(j = i+1; j<N; j++)
            if(strcmp(DS[i],DS[j]) > 0){
                strcpy(s,DS[i]);
                strcpy(DS[i],DS[j]);
                strcpy(DS[j],s);
            }
    //In danh sach da sap xep
    printf("\nDS sinh vien vua nhap:");
    for(i = 0; i<N; i++)
        printf("\n%s",DS[i]);
    return 0;
}
```

```
So sinh vien: 3
Ten sinh vien [1]: Tran Mai
Ten sinh vien [2]: Hoang Hoa
Ten sinh vien [3]: Nguyen Ngan

DS sinh vien vua nhap:
Hoang Hoa
Nguyen Ngan
Tran Mai
-----
```

1. Nhập vào 2 chuỗi S1, S2 và một số nguyên k. Hãy chèn chuỗi S1 vào S2 và đưa ra màn hình (*giả thiết chuỗi S2 được khai báo đủ lớn*)
2. Một văn bản gồm không quá 30 dòng, mỗi dòng không quá 50 ký tự. Hãy viết chương trình thực hiện nhập vào một văn bản, sau đó
  - a. Nhập vào chuỗi s và chỉ ra vị trí xuất hiện của chuỗi S trong văn bản **nếu có**.
  - b. Thay tất cả các chuỗi "hanoi" (nếu có) bằng chuỗi "HANOI"
  - c. Đếm xem trong văn bản có bao nhiêu từ (*các từ phân cách bởi dấu cách*)
  - d. Tính tần suất xuất hiện của các từ trong văn bản