TRƯỜNG ĐẠI HỌC KINH TẾ KỸ THUẬT CÔNG NGHIỆP
Khoa Công Nghệ Thông Tin

FUTURE TECHNOLOGY
CÔNG NGHỆ TƯƠNG LAI

# Phần 2. NN lập trình C++

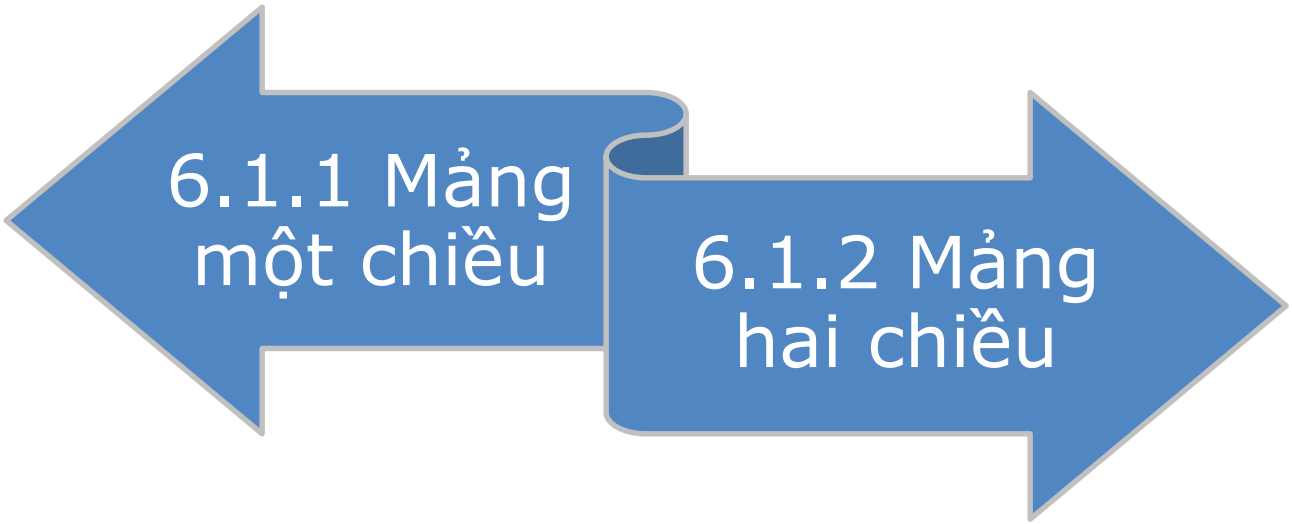
# CHƯƠNG 6
# MẢNG VÀ XÂU KÝ TỰ

# Chương 6 – Mảng và xâu ký tự

## 6.1 Mảng

## 6.2 Xâu ký tự

# 6.1 Mảng



# 6.1.2 Mảng hai chiều

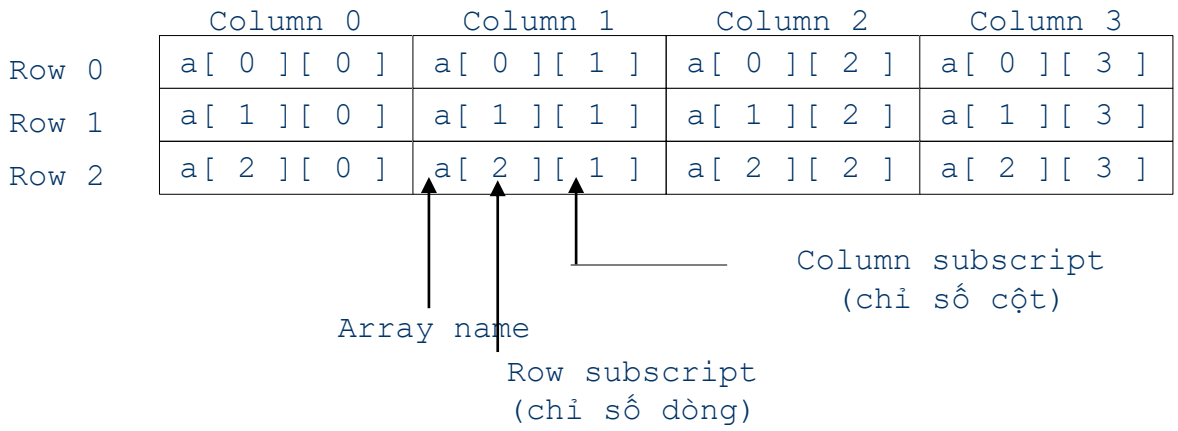
- Cú pháp:

**<*Kieu_dl*>  <*ten_mang*>[**so_hang**][**so_cot**];**

- Phần tử của mảng có kiểu dữ liệu cụ thể
- Một mảng có thể được coi là kiểu dữ liệu cho phần tử của mảng.
  - Một phần tử của mảng có thể là một mảng khác
- Mảng 2 chiều là: “mảng của các mảng”

# 6.1.2 Mảng hai chiều

- Truy cập mảng
  - int a[ 3 ][ 4 ];
  - **a[i][j]**
  - Truy cập thông qua tên mảng cùng chỉ số hàng và cột
  - "Mảng của mảng"
    - **a[0]** là một mảng 4 phần tử
    - **a[0][0]** là phần tử đầu tiên của mảng



# 6.1.2 Mảng hai chiều

- Khởi tạo
  - Mặc định là 0
  - Khởi tạo, mỗi dòng trong 1 cặp ngoặc

```
int b[2][2] = { { 1, 2 }, { 3, 4 } };
```

Row 0 Row 1

| | |
|---|---|
| 1 | 2 |
| 3 | 4 |

```
int b[2][2] = { { 1 }, { 3, 4 } };
```

| | |
|---|---|
| 1 | 0 |
| 3 | 4 |

# 6.1.2 Mảng hai chiều

- Nhập mảng

duyệt qua tất cả các phần tử

```
int a[3][4];
for(int i = 0; i < 3; i++)
  for(int j = 0;  j < 4; j++)
  {
     cout<<"a[ "<<i<<"]["<<j<<"]= ";
     cin>>a[i][j];
     }
```

nhập dữ liệu cho a[i][j]

# 6.1.2 Mảng hai chiều

- Xuất mảng

```
int a[3][4];
for(int i = 0; i < 3; i++)
{    for(int j = 0; j < 4; j++)
          cout<<a[i][j]<<"\t";
     cout<<endl;
}
```

# Chương 6 – Mảng và xâu ký tự

## 6.1 Mảng

## 6.2 Xâu ký tự

# 6.2 Xâu ký tự

- Khái niệm
  - Kiểu char chỉ chứa được một ký tự. Để lưu trữ một xâu (nhiều ký tự) ta sử dụng mảng (một chiều) các ký tự.
  - Xâu ký tự kết thúc bằng ký tự '\0' (null)

    ➔ Độ dài xâu = kích thước mảng – 1
- Ví dụ

```
char hoten[30];      // Dài 29 ký tự
char ngaysinh[9];    // Dài 8 ký tự
```

# 6.2 Xâu ký tự

- Khai báo
  - char <ten_xau>[do_dai_xau];

- Ví dụ

```
char hoten[30];
char ngaysinh[9];
```

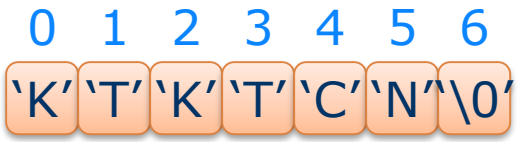
# 6.2 Xâu ký tự

- Khởi tạo như mảng thông thường
  - Độ dài cụ thể

```
char s[10] = {'K', 'T', 'K', 'T', 'C', ' N', '\0'};
char s[10] = "KTKTCN";      // Tự động thêm '\0'
```


  - Tự xác định độ dài

```
char s[] = {'K', 'T', 'K', 'T', 'C', 'N', '\0'};
char s[] = "KTKTCN";// Tự động thêm '\0'
```



# 6.2 Xâu ký tự

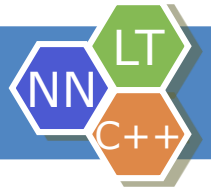
- Nhập xâu **cin.getline**

```
char st[10];
cin.getline(st,10);
```

  - Ghi dữ liệu vào của người dùng vào xâu
    - Dừng lại ở ký tự trắng đầu tiên (tab, newline, blank…)
    - Thêm vào ký tự **null**
  - Nếu nhập quá nhiều, dữ liệu sẽ tràn mảng
    - Ta cần phải tránh điều này (mục 5.12 sẽ giải thích phương pháp)
- Xuất xâu

```
cout << st << endl;
```

  - Không sử dụng được với các mảng có kiểu dữ liệu khác
  - In các ký tự cho đến khi gặp **null**




# Một số hàm thao tác trên xâu

- Thuộc thư viện <string.h>
  - strlen: độ dài của xâu
  - strcpy: sao chép xâu
  - strlwr/strupr: biến thành chữ thường/chữ hoa
  - strcmp: so sánh hai xâu
  - strcat: nối hai xâu
  - strstr: tìm vị trí của xâu này trong xâu kia
  - strdup: tạo bản sao của xâu cho trước
  - strrev: đảo ngược xâu

# Hàm tính độ dài xâu

**size_t* strlen(const char *s)**

Tính độ dài xâu s

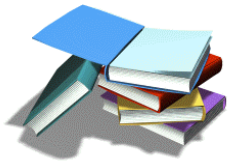
size_t thay cho unsigned (trong <stddef.h>) dùng để đo các đại lượng không dấu.


Độ dài xâu s

```
char s[] = "Tin hoc co so A!!!";
int len = strlen(s);            // => 18
```

# Hàm sao chép xâu

**char *strcpy(char dest[], const char src[])**



Sao chép xâu src sang xâu dest, dừng khi ký tự kết thúc xâu '\0' vừa được chép.
! dest phải đủ lớn để chứa src

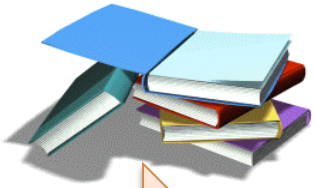
Trả về

Địa chỉ xâu dest

```
char s[100];
s = "Tin hoc co so A";          // sai
strcpy(s, "Tin hoc co so A");   // đúng
```

# Hàm đảo ngược xâu

**char *strrev(char *s)**



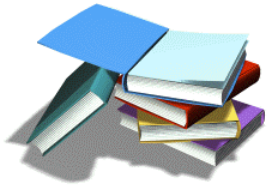
Đảo ngược thứ tự các ký tự trong xâu (trừ ký tự kết thúc xâu)


- Địa chỉ xâu kết quả

```
char s[] = "Tin hoc co so A!!!";
strrev(s);
puts(s);                    // !!!A os oc coh  niT
```

# Hàm so sánh hai xâu

**int strcmp(const char *s1, const char *s2)**



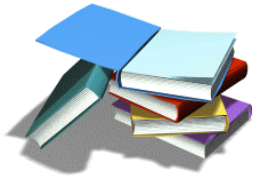
So sánh hai xâu s1 và s2 (phân biệt hoa thường)


- < 0 nếu s1 < s2
- == 0 nếu s1 == s2
- >0 nếu s1 > s2

```
char s1[] = "tin hoc co so A!!!";
char s2[] = "hoc tin co so A!!!";
int kq = strcmp(s1, s2);     // => kq > 0
```

# Hàm so sánh hai xâu

**int stricmp(const char *s1, const char *s2)**



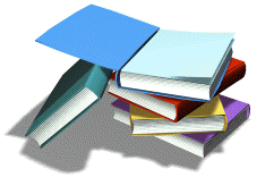
So sánh hai xâu s1 và s2 (không phân biệt hoa thường)


- < 0 nếu s1 < s2
- == 0 nếu s1 == s2
- >0 nếu s1 > s2

```
char s1[] = "tin hoc co so A!!!";
char s2[] = "TIN HOC CO SO A!!!";
int kq = stricmp(s1, s2);    // => kq == 0
```

# Hàm nối hai xâu

**char* strcat(char *dest, const char *src)**



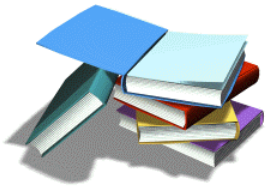
Nối xâu src vào sau xâu dest.
! xâu dest phải đủ chứa kết quả


- Địa chỉ của xâu được nối

```
char s1[100] = "Tin hoc";
char s2[] = "co so A!!!";
strcat(s1, " ");      // => "Tin hoc "
strcat(s1, s2);       // => "Tin hoc co so A!!!"
```

# Hàm tìm xâu trong xâu

**char* strstr(const char *s1, const char *s2)**



Tìm vị trí xuất hiện đầu tiên của s2 trong s1


- Thành công: trả về con trỏ đến vị trí xuất hiện đầu tiên của s2 trong s1.
- Thất bại: trả về null

```
char s1[] = "Tin hoc co so A!!!";
char s2[] = "hoc";
if (strstr(s1, s2) != null)
        cout<<"Tim thay!";
```

# Câu hỏi củng cố bài

1. Hàm thư viện được sử dụng để đảo ngược xâu?

   A. strcpy

   B. strcmp

   C. strcat

   D. strrev

Multiple Choice

# Câu hỏi củng cố bài

2. Hàm thư viện được sử dụng để chuyển đổi thành xâu chữ thường?

A. strupr

B. strlwr

C. strcat

D. strrev

Multiple Choice

# Câu hỏi củng cố bài

3. Đoạn lệnh sau cho kết quả thế nào?

```
#include <iostream.h>
#include <conio.h>
int main()
{
    char str[] = "VNCODING\0\.NET\0";
    cout<< str;
    return 0;
}
```

A. VNCODING\0

B. VNCODING

C. VNCODING\0.NET

D. VNCODING\0.NET\0

Multiple Choice

# Câu hỏi củng cố bài

4. Đoạn lệnh sau cho kết quả thế nào?

A. pvqit

B. pvpit

C. pvrit

D. vrit

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

int main()
{
    char str[] = {"pvpit"};
    str[2]++;
    cout<<str;
}
```

Multiple Choice

# Câu hỏi củng cố bài

5. Đoạn lệnh sau cho kết quả thế nào?

A. 7

B. 6

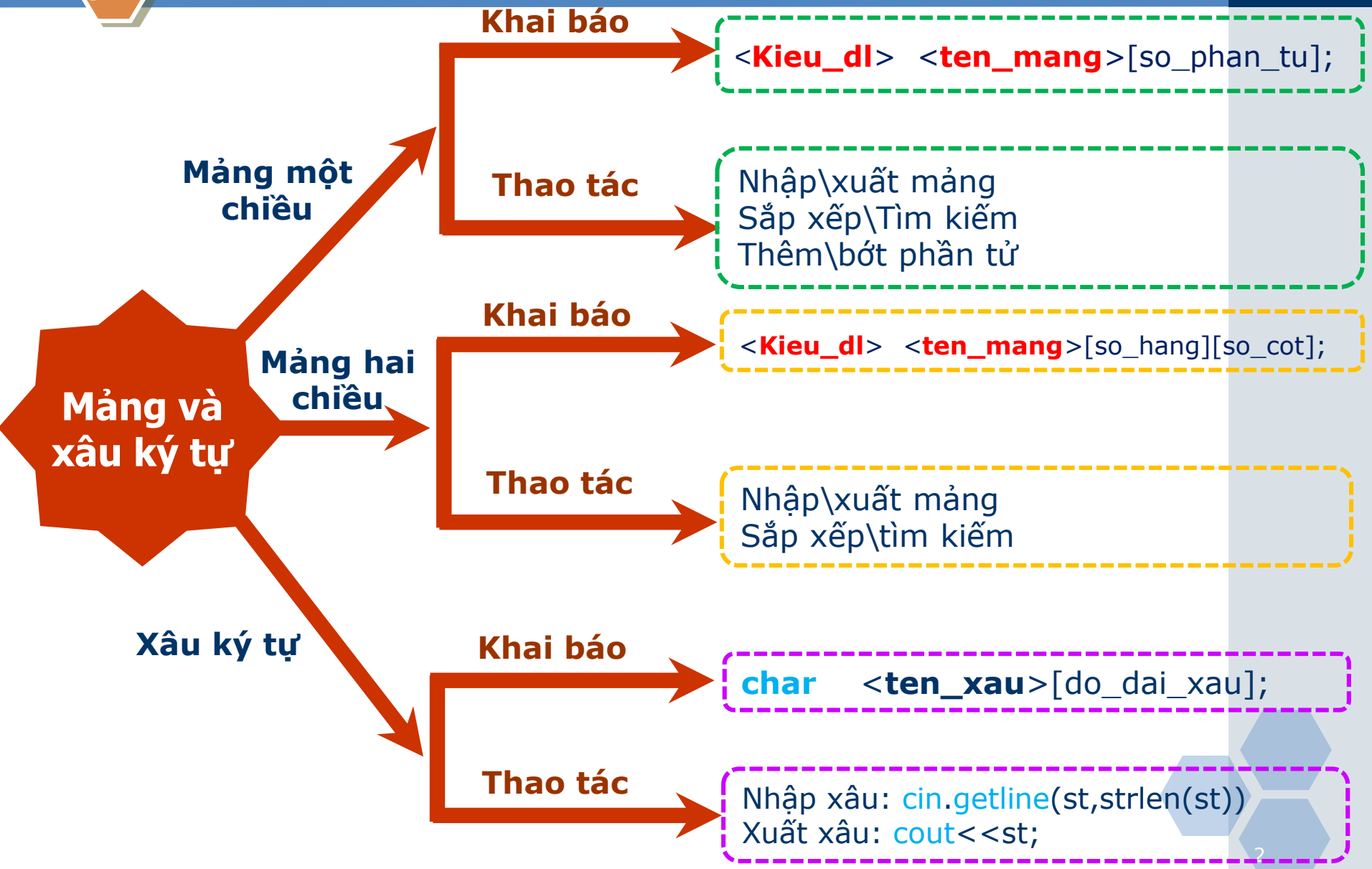
C. 5

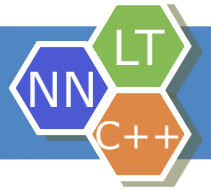
D. 4

```
1 #include <iostream.h>
2 #include <string.h>
3
4 int main()
5 {
6     char s[] = "\12345s\n";
7     cout<< strlen(s);
8
9 }
```

Multiple Choice

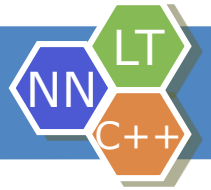
# Tổng kết


Mảng và xâu ký tự
Mảng một chiều
Khai báo
<Kieu_dl> <ten_mang>[so_phan_tu];
Thao tác
Nhập\xuất mảng
Sắp xếp\Tìm kiếm
Thêm\bớt phần tử
Mảng hai chiều
Khai báo
<Kieu_dl> <ten_mang>[so_hang][so_cot];
Thao tác
Nhập\xuất mảng
Sắp xếp\tìm kiếm
Xâu ký tự
Khai báo
char <ten_xau>[do_dai_xau];
Thao tác
Nhập xâu: cin.getline(st,strlen(st))
Xuất xâu: cout<<st;




# Câu hỏi lý thuyết

1. Nêu định nghĩa xâu. Cách khai báo xâu ký tự
2. Nêu cách nhập, xuất xâu.
3. Nêu cú pháp các hàm thao tác trên xâu.




# Bài tập

1. Viết chương trình nhập vào một ma trận m hàng, n cột các phần tử thực.

a) Tính trung bình cộng các phần tử âm của mảng

b) Tìm phần tử lớn nhất trong mảng

2. Viết chương trình nhập vào một ma trận vuông cấp n các phần tử nguyên.

a) Tính tổng các phần tử trên đường chéo chính của ma trận

b) Tính tổng các phần tử trên đường chéo phụ của ma trận

# Bài tập

3. Viết chương trình C++ thực hiện các công việc sau:

- Nhập vào xâu s, đếm số từ trong xâu?
- In xâu đảo ngược của xâu s?
- Đếm số lượng ký tự là a,b trên xâu?

# Bài tập

4. Viết chương trình C++ thực hiện các công việc sau:

- Nhập vào một xâu ký tự
- Đếm tần suất xuất hiện của mỗi ký tự trên xâu

Ví dụ: “lelan”  L xuat hien 2 lan, e xuat hien 1 lần, a xuất hiện 1 lần, n xuất hiện 1 lần