

Chương 7: chuỗi (string)

1. Định nghĩa

- Chuỗi là một mảng các ký tự với ký tự null ở cuối chuỗi. Ký tự null (`\0`) là ký tự dùng để kết thúc chuỗi.
- Chuỗi ký tự được đặt trong ngoặc kép.
- Ví dụ: “This is a string.”

1. Định nghĩa

- Trong C++, có 2 loại kiểu chuỗi, kiểu chuỗi C truyền thống (C-style string) và kiểu chuỗi C++ (C++ - style string). Trong bài học này sẽ chỉ tập trung về chuỗi kiểu C.

2. Khai báo chuỗi

- CÁCH 1: Khai báo chuỗi là 1 mảng 1 chiều.
 - Cú pháp `char strName[strLength];`
 - **strName**: Tên chuỗi
 - **strLength**: độ dài của chuỗi
 - Ví dụ: Viết lệnh khai báo chuỗi có 50 ký tự
 - `char name[50];`

2. Khai báo chuỗi

- CÁCH 2 : Dùng con trỏ

- Cú pháp:

```
char *strName;
```

- **strName**: Tên chuỗi

- Ví dụ:

- char *name;

- name = new char[51];

3. Khởi tạo chuỗi

- Có 2 cách để nhập dữ liệu cho chuỗi
 - Cách 1: dùng hàm gets()

```
char *gets(char *s);
```

- Ví dụ:

```
#include <iostream.h>
#include <stdio.h>
void main()
{
    char strName[10];
    gets(strName);
}
```

3. Khởi tạo chuỗi

- Cách 2: dùng lệnh **cin**
- Ví dụ

```
#include <iostream.h>
void main()
{
    char strName[100];
    cin>>strName;
}
```

3. Khởi tạo chuỗi

- **Ghi chú**: Tuy nhiên, khi dùng `cin>>` để nhập dữ liệu cho chuỗi, chương trình sẽ tự động ngắt chuỗi khi gặp ký tự khoảng trắng trong chuỗi → Để chuỗi không bị ngắt khi gặp ký tự khoảng trắng, ta sẽ dùng hàm `cin.getline()`
- `cin.getline(strname, strlen)`

3. Xuất chuỗi

- Có 2 cách để nhập dữ liệu cho chuỗi
 - Cách 1: dùng hàm puts()

```
int puts(const char *s);
```

- Ví dụ:

```
#include <iostream.h>
#include <stdio.h>
void main()
{
    char strName[10];
    gets(strName);
    puts(strName);
}
```

4. Xuất chuỗi

- Cách 2: dùng lệnh **cout**
- Ví dụ:

```
#include <iostream.h>
void main()
{
    char strName[10];
    cin>>strName;
    cout<<strName;
}
```

5. Một số hàm thư viện thao tác trên chuỗi

- Các hàm thư viện thao tác trên chuỗi nằm trong thư viện **<string.h>**
- Một số hàm phổ biến

Tên hàm	Chức năng
strcpy(s1, s2)	Sao chép chuỗi s2 vào s1
strcat(s1, s2)	Nối chuỗi s2 vào cuối chuỗi s1
strlen(s1)	Trả về độ dài của chuỗi s1
strcmp(s1, s2)	So sánh 2 chuỗi s1 và s2. Nếu s1 = s2 => kết quả trả về = 0 Nếu s1 < s2 => kết quả trả về < 0 Nếu s1 > s2 => kết quả trả về > 0
strchr(s1,ch)	Trả về vị trí đầu tiên của ký tự ch trong s1
strstr(s1,s2)	Trả về vị trí đầu tiên của s2 trong s1

5. Một số hàm thư viện thao tác trên chuỗi

- Ví dụ:

```
#include <iostream.h>
#include <cstring.h>
int main()
{
    char name[50] , lastname[50], fullname[100];
    cout<<"Please enter your name: ";
    cin.getline ( name, 50 );
    if ( strcmp ( name, "Julienne" ) == 0 )
        cout<<"That's my name too.\n";
    else // Not equal
        cout<<"That's not my name.\n";
    cout<<"Your name is "<< strlen ( name ) <<" letters long\n";
    cout<<"Enter your last name: "; cin.getline ( lastname, 50 );
    fullname[0] = '\0';
    strcat ( fullname, name ); // Copy name into full name
    strcat ( fullname, " " );
    strcat ( fullname, lastname );
    cout<<"Your full name is "<< fullname <<"\n";
    cin.get(); }
```

6. Mảng các chuỗi

- Mảng các chuỗi là 1 mảng 2 chiều. Kích thước của chỉ số thứ nhất là số chuỗi và chỉ số thứ 2 là độ dài của mỗi chuỗi.
- Ví dụ: `char s[5][50];`
- Để nhập dữ liệu cho 1 chuỗi thứ *i* trong mảng, ta dùng lệnh
 - `cin>>s[1];`

6. Mảng các chuỗi

- Khai báo và khởi tạo mảng các chuỗi.

```
char arrayList[][length] = {  
    constantString1,  
    constantString2,  
    ...  
    constantStringN};
```

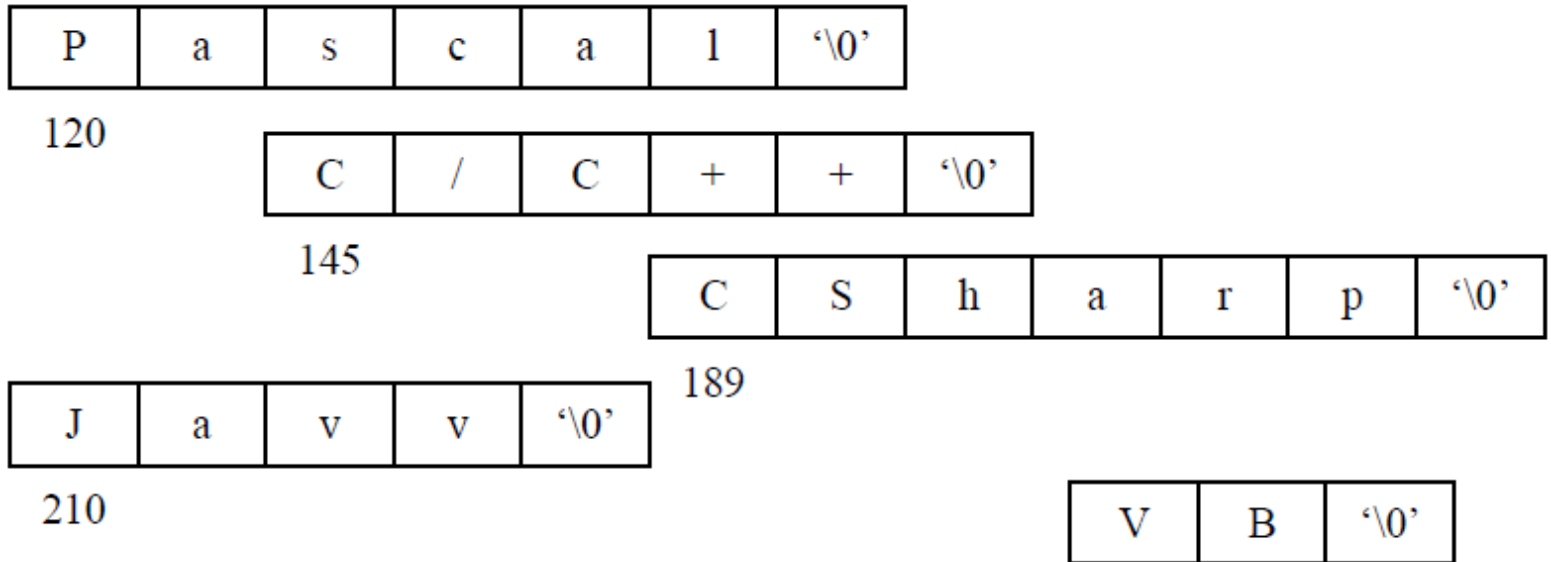
- Ví dụ:

```
char listOfPL[][10] = {"Pascal", "C/C++",  
"CSharp", "Java", "VB"};
```

6. Mảng con trỏ trỏ đến các chuỗi

- Ngoài cách dùng mảng ký tự 2 chiều để lưu trữ mảng các chuỗi, ta có thể dùng mảng các con trỏ, mỗi con trỏ sẽ giữ địa chỉ của 1 chuỗi.
- Ví dụ: `char *str[20];`
- `char *listOfPL[] = {"Pascal", "C/C++", "CSharp", "Java", "VB"};`

6. Mảng con trỏ trỏ đến các chuỗi



Giá trị	120	145	189	210	272	
Địa chỉ bộ nhớ	65514	65516	65518	65520	65522	

