

# KỸ THUẬT LẬP TRÌNH NGÀNH ĐIỆN TỬ - VIỄN THÔNG

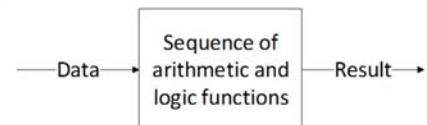
Giảng viên : CAO TRẦN BẢO THƯƠNG  
Bộ môn MÁY TÍNH – HỆ THỐNG NHÚNG  
Khoa Điện Tử - Viễn Thông  
Đại Học Khoa Học Tự Nhiên – TP.HCM

1

## Phần 1 : Tổng quan

### Phần cứng vs Máy tính

- ★ **Phần cứng (ASIC – Application - Specific Integrated Circuit):** chức năng cố định, tốc độ cao, không lập trình được

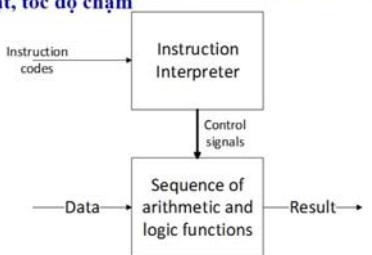


2

## Phần 1 : Tổng quan

### Phần cứng vs Máy tính

#### Máy tính: có khả năng lập trình (tập lệnh), chức năng linh hoạt, tốc độ chậm



3

### Hệ thống máy tính: có 2 loại chính

#### Máy tính cá nhân (Personal computer)

#### Đa nhiệm

Ví dụ: laptop, desktop, smartphone

#### Hệ thống nhúng (Embedded System)

#### Đơn nhiệm

Ví dụ: máy chụp hình, máy chiếu...

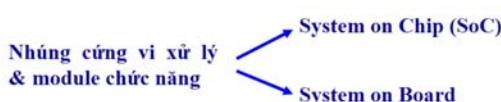
4

So sánh tốc độ xử lý, công suất tiêu thụ, chi phí giữa PC và hệ thống nhúng?

## Phần 1 : Tổng quan

### Hệ thống nhúng (Embedded System)

#### Hệ thống nhúng trên nền tảng ASIC



Board có giá thành thấp, tốc độ cao, công suất thấp, chức năng cố định

5

### Hệ thống nhúng (Embedded System)

#### Hệ thống nhúng trên nền tảng FPGA

Vì xử lý và các module chức năng có thể nhúng cứng hoặc mềm

Board có giá thành cao, tốc độ chậm, công suất cao, linh hoạt

6

## Phản 1 : Tổng quan



### Hệ thống nhúng (Embedded System)

#### ★ SoPC (System on Programmable Chip)

Là một hệ thống nhúng được tích hợp vào trong một con chip đơn duy nhất. Với nền tảng FPGA của hãng Altera, ta có thể xây dựng một SoPC với vi xử lý Nios II.

Vi xử lý Nios II là một dạng vi xử lý soft-core sử dụng chính tài nguyên của FPGA tạo thành.

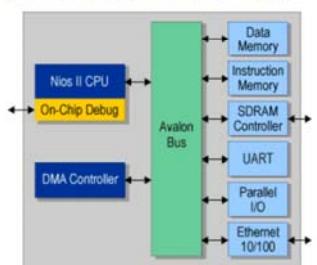
7  
ctruyenthong

## Phản 1 : Tổng quan



### Hệ thống nhúng (Embedded System)

#### ★ SoPC (System on Programmable Chip)



9  
ctruyenthong

## Phản 1 : Tổng quan



### Một số khái niệm về lập trình

★ Chương trình (program) gồm một dãy các lệnh/chỉ thị (instruction) liên tiếp nhau điều khiển sự hoạt động của hệ thống máy tính

★ Lập trình viên (programmer) là người viết chương trình

★ Tập tin thực thi (executable program file) là tập tin chứa các lệnh cho CPU thực hiện. Trong hệ điều hành Windows, tập tin có dạng \*.exe

11  
ctruyenthong

## Phản 1 : Tổng quan



### Hệ thống nhúng (Embedded System)

#### ★ SoPC (System on Programmable Chip)

Một số CPU nhúng khác: ARM, MIPS, Coldfire/68k, PowerPC, PIC, Atmel AVR, Renesas H8, SH, V850, FR-V, M32R, Z80, Z8 ...

8  
ctruyenthong

## Phản 1 : Tổng quan



### Hệ thống nhúng (Embedded System)

#### ★ SoPC (System on Programmable Chip)

Phần cứng

Vi xử lý Nios II

Bộ nhớ

Các thành phần ngoại vi...

Công cụ: Quartus II

Ngôn ngữ lập trình: Verilog HDL

Phần mềm

Công cụ: Nios II IDE

Ngôn ngữ lập trình : C HDL

10  
ctruyenthong

## Phản 1 : Tổng quan



### Một số khái niệm về lập trình

## Phản 1 : Tổng quan

★ Người sử dụng/ người dùng cuối (end user) là người thực hiện chương trình do lập trình viên viết sẵn

★ Chương trình mã máy (machine code program) là dạng cơ bản nhất của chương trình chỉ bao gồm các bit nhị phân, có thể nạp trực tiếp vào CPU để thực hiện

★ Hợp ngữ (assembly language) là ngôn ngữ lập trình trừu tượng hơn mã máy, mỗi lệnh sẽ có một từ thay thế cho chuỗi bit của mã máy

12  
ctruyenthong

## Phần 1 : Tổng quan



### Một số khái niệm về lập trình

- ★ Trình hợp dịch (assembler) dùng để chuyển hợp ngữ ra mã máy với mỗi lệnh trong hợp ngữ tương ứng một lệnh mã máy
- ★ Ngôn ngữ lập trình (programming language) là ngôn ngữ để lập trình viên viết chương trình cho máy tính. Hợp ngữ cũng là một ngôn ngữ lập trình cấp thấp (low-level programming language)

13

## Phần 1 : Tổng quan



### Một số khái niệm về lập trình

- ★ Ngôn ngữ Java, C# là ngôn ngữ lập trình hiện đại và chuyên nghiệp, đặc biệt Java được sử dụng để lập trình trên nền tảng hệ điều hành Android
- ★ Ngôn ngữ PHP được dùng trong các ứng dụng web, quản trị mạng
- ★ Ngôn ngữ Python là lập trình hướng đối tượng, cấp cao, mạnh mẽ, dễ dàng tiếp cận, tương thích đa nền tảng

15

## Phần 1 : Tổng quan



### Một số khái niệm về lập trình

- ★ Môi trường phát triển chương trình tích hợp (IDE\_Integrated Development Environment) là công cụ do các hãng phần mềm cung cấp nhằm hỗ trợ lập trình viên
- ★ Một số IDE tiêu biểu như Visual Studio, Jbuilder, Eclipse, Nios II IDE...

17

## Phần 1 : Tổng quan



### Một số khái niệm về lập trình

- ★ Trình dịch cấp cao dùng để chuyển ngôn ngữ lập trình cấp cao sang mã máy. Có 2 loại chính là trình biên dịch (compiler) và trình thông dịch (interpreter)
- ★ Trình biên dịch có kết quả cuối cùng là tập tin thực thi được lưu vào bộ nhớ ngoài
- ★ Trình thông dịch chuyển ngôn ngữ cấp cao sang mã máy, nạp trực tiếp vào bộ nhớ lệnh để thực thi ngay

14

## Phần 1 : Tổng quan



### Một số khái niệm về lập trình

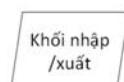
- ★ Lưu đồ (Flow chart) giải thuật là phương tiện diễn giải thuật toán một cách trực quan. Lưu đồ được tạo bởi các khối nối với nhau.

## Phần 1 : Tổng quan



### Một số khái niệm về lập trình

- ★ Lưu đồ (Flow chart) giải thuật là phương tiện diễn giải thuật toán một cách trực quan. Lưu đồ được tạo bởi các khối nối với nhau.

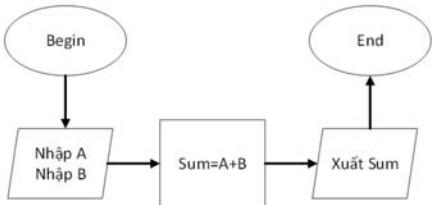


18

## Phần 1 : Tổng quan

### Một số khái niệm về lập trình

- Ví dụ: vẽ lưu đồ giải thuật cho chương trình tính tổng 2 số A và B



19

## Phần 1 : Tổng quan

### Một số khái niệm về lập trình

- Mã giả (Pseudocode) dùng để diễn giải thuật toán gần với ngôn ngữ lập trình hơn. Mã giả không hề theo cấu trúc cú pháp của bất kỳ ngôn ngữ lập trình cụ thể nào.

- Ví dụ: viết mã giả cho chương trình tính tổng 2 số A và B

Begin

Nhập A, B

Sum=A+B

Xuất Sum

End

20

## Phần 2 : Dữ liệu cơ sở và phép toán

### Ngôn ngữ lập trình C

- C ra đời tại phòng thí nghiệm Bell Telephone thuộc tập đoàn AT&T giữa những năm 1969 và 1973
- C là ngôn ngữ lập trình hệ thống, firmware, driver, cơ sở dữ liệu
- ANSI C là ngôn ngữ C chuẩn ra đời vào năm 1989

21

## Phần 2 : Dữ liệu cơ sở và phép toán

### Chương trình đầu tiên

```
1 // in ra chữ Hello World !
2 #include "StdAfx.h"
3 #include <stdio.h>
4 #include <conio.h>
5 void main()
6 {
7     printf ("Hello World !");
8 }
9 /*Chạy (run) trên máy tính và xem kết quả hiển thị lên console*/
```

22

## Phần 2 : Dữ liệu cơ sở và phép toán

### Chương trình đầu tiên

- Dòng 1: // <nội dung> để ghi chú trên 1 dòng
- Dòng 9: /\* <nội dung> \*/ để ghi chú trên nhiều dòng
- Dòng 2, 3, 4: khai báo sử dụng các thư viện chuẩn
- Dòng 5: chương trình chính với hàm main
- Dòng 7: in ra câu chữ bất kỳ bằng lệnh printf
- Dòng 6, 8: bắt đầu/ kết thúc chương trình

23

## Phần 2 : Dữ liệu cơ sở và phép toán

### Bài tập

- Viết chương trình hiển thị như sau:

Khoa Dien Tu – Vien Thong

Chuyen nganh May Tinh – He Thong Nhung

Ho va Ten: &lt;tên sv&gt;

Ma so: &lt;mã số sv&gt;

Với \n dùng để xuống hàng, \t dùng để thụt vào 1 tab

Lệnh getch(); để dừng màn hình lại xem

24

## Phần 2 : Dữ liệu cơ sở và phép toán



## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

```

1 // Bài tập
2 #include "StdAfx.h"
3 #include <stdio.h>
4 #include <conio.h>
5 void main()
6 {
7     .....
8     .....
9     .....
10    .....
11 }

```

25

chu\_2013\_javas

### Kiểu dữ liệu, hằng, biến

★ Các kiểu dữ liệu cơ bản trong C gồm có

+ Kiểu số nguyên

char	1 byte	%c (ký tự), %d (số)
int	2 byte	%d
long	4 byte	%d

Kiểu số nguyên có thể thêm unsigned để biểu diễn số không dấu

26

chu\_2013\_javas

## Phần 2 : Dữ liệu cơ sở và phép toán



## Phần 2 : Dữ liệu cơ sở và phép toán



### Kiểu dữ liệu, hằng, biến

★ Các kiểu dữ liệu cơ bản trong C gồm có

+ Kiểu số thực

float	4 byte	%f (%.2f làm tròn)
double	8 byte	%lf
long double	10 byte	%lf

Có thể xem kích thước kiểu dữ liệu (theo byte) bằng lệnh sizeof(<kiểu dữ liệu>). Ví dụ:

```
printf ("Kich thuoc kieu float=%d",sizeof(float));
```

27

chu\_2013\_javas

### Kiểu dữ liệu, hằng, biến

★ Khai báo hằng số

Hằng số được sử dụng trong chương trình mà không thay đổi

Ví dụ: khai báo hằng số Pi

```
#define Pi 3.14159
```

28

chu\_2013\_javas

## Phần 2 : Dữ liệu cơ sở và phép toán



## Phần 2 : Dữ liệu cơ sở và phép toán



### Kiểu dữ liệu, hằng, biến

★ Khai báo biến

Cách đặt tên biến và hằng phải dùng các chữ cái từ A đến Z, chữ số từ 0 đến 9, dấu gạch dưới “\_”

Tên biến/hằng không được trùng với từ khóa

Tên biến/hằng phải bắt đầu bằng chữ cái

Tên phải có tính gợi nhớ đến đối tượng

Ví dụ: Dien\_tich, DienTich, KetQua\_1, Ket\_Qua2

29

chu\_2013\_javas

### Kiểu dữ liệu, hằng, biến

★ Khai báo biến

Một số khai báo KHÔNG hợp lệ

So Luong // có khoảng trắng

35a, 1\_X // bắt đầu bằng số

KetQua@ // ký tự không hợp lệ

char, default // trùng từ khóa

30

chu\_2013\_javas

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

★ Viết chương trình nhập tên của bạn. Sau đó xuất ra thông báo:

Khoa Dien Tu – Vien Thong chao ban <ten>

Chuc ban <ten> tot nghiep thu khoa

Khai báo chuỗi sử dụng kiểu char. Ví dụ:

char TenSV[80]; // chuỗi có 80 ký tự

31

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

★ Kết quả

```
Nhap ten:Nguyen Van A  
Khoa Dien Tu - Vien Thong chao ban Nguyen Van A  
Chuc ban Nguyen Van A tot nghiep thu khoa.
```

33

## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

★ Nhập dữ liệu

Thông thường sử dụng lệnh scanf

Ví dụ nhập chiều dài của hình chữ nhật với biến Dai  
kiểu int

scanf("%d", &Dai);

Ví dụ nhập bán kính của hình tròn với biến R kiểu float

scanf("%f", &R);

35

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

```
1 <khai báo thư viện>  
2 void main()  
3 {  
4     char TenSV[80];  
5     printf ("Nhap ten:");  
6     .....  
7     .....  
8     .....  
9     .....  
10    getch();  
11 }
```

32

## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

★ Các chương trình cơ bản đều có 3 bước chính:

Nhập dữ liệu từ người sử dụng

Tính toán/xử lý dữ liệu để ra kết quả mong muốn

Xuất dữ liệu sau khi đã tính toán ra màn hình (hoặc thiết bị tương ứng)

34

## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

★ Nhập dữ liệu

Thông thường sử dụng lệnh scanf

Ví dụ nhập chiều dài của hình chữ nhật với biến Dai  
kiểu int

scanf("%d", &Dai);

Ví dụ nhập bán kính của hình tròn với biến R kiểu float

scanf("%f", &R);

35

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

★ Cho biết lệnh nào đúng/sai với biến là số nguyên

- 1/ scanf ("%d",A);
- 2/ scanf ("%5d",&A);
- 3/ scanf ("%d",&B);
- 4/ scanf ("%d%d",A,B);
- 5/ scanf ("%d",&C,&D);
- 6/ scanf ("%d%d",&E,&F);

36

## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

#### Xuất dữ liệu

Thông thường sử dụng lệnh printf

Ví dụ xuất chiều dài của hình chữ nhật với biến Dai kiểu int

```
printf("Chieu dai la %d",Dai);
```

Ví dụ xuất bán kính của hình tròn với biến R kiểu float

```
printf("Ban kinh la %f",R);
```

37

## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

#### Tính toán: Toán tử số học

```
char A= 10; char B = 3;
```

Cho biết kết quả của phép gán :

1/ out = (A + B) ; // out = ?

2/ out = (B \* A) ; // out = ?

3/ out = (A / B) ; // out = ?

4/ out = (A % B) ; // out = ?

5/ out = (B % A) ; // out = ?

39

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

```

1 <khai báo thư viện>
2 #define Pi 3.1415      //khai báo hằng số Pi
3 void main()
4 {
5     float R;
6     float DienTich;
7     printf ("Nhập bán kính R:");
8     scanf ("%f",&R);
9
10    printf ("Dien tich hinh tron voi ban kinh %f
11 la: %f",R,DienTich);
12    getch();
13 }
```

41

## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

#### Tính toán: Toán tử số học

Ký hiệu	Chức năng	# Toán hạng
+	Cộng	2
-	Trừ	2
*	Nhân	2
/	Chia	2
%	Lấy phần dư	2
-	Số âm	1

38

## Phần 2 : Dữ liệu cơ sở và phép toán



## Phần 2 : Dữ liệu cơ sở và phép toán

### Bài tập

#### Viết chương trình nhập vào bán kính hình tròn. Sau đó xuất ra diện tích hình tròn

40

## Phần 2 : Dữ liệu cơ sở và phép toán



## Phần 2 : Dữ liệu cơ sở và phép toán

### Bài tập

#### Viết chương trình nhập vào chiều dài, chiều rộng hình chữ nhật theo số nguyên. Sau đó xuất ra chu vi, diện tích hình chữ nhật.

42

## Phần 2 : Dữ liệu cơ sở và phép toán

### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     int Dai, Rong, ChuVi, DienTich;
5     printf ("Nhập chiều dài:");
6     scanf ("%d",&Dai);
7     printf ("Nhập chiều rộng:");
8     scanf ("%d",&Rong);
9
10
11
12
13 }
```

43

## Phần 2 : Dữ liệu cơ sở và phép toán

### Bài tập

★ Viết chương trình nhập vào chiều dài, chiều rộng hình chữ nhật. Sau đó xuất ra đường chéo hình chữ nhật.

## Phần 2 : Dữ liệu cơ sở và phép toán

### Bài tập

★ Kết quả

```
Nhập chiều dài:10
Nhập chiều rộng:8
Chu vi là 36
Diện tích là 80
Chiều dài đường chéo là 12.81
```

45

## Phần 2 : Dữ liệu cơ sở và phép toán

### Bài tập

★ Viết chương trình nhập vào một chữ số nguyên có tối đa 3 chữ số. Sau đó xuất ra giá trị hàng trăm, hàng chục, hàng đơn vị của số này.

## Phần 2 : Dữ liệu cơ sở và phép toán

### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     int SoNguyen, Tram, Chuc, DonVi;
5     printf ("Nhập số nguyên tối đa 3 chữ số:");
6     scanf ("%d",&SoNguyen);
7
8
9
10
11
12
13
14
15 }
```

47

## Phần 2 : Dữ liệu cơ sở và phép toán

### Bài tập

★ Viết chương trình nhập vào số giây bất kỳ. Sau đó xuất ra giá trị giờ phút giây tương ứng.

Ví dụ: 3661 giây <-> 1 giờ 1 phút 1 giây

48



## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

★ Tính toán: Toán tử logic đại số Boolean

```
char A= 11;  char B = 0;
```

Cho biết kết quả của phép gán :

```
1/ out = A && B ;           // out = ?  
2/ out = !A ;               // out = ?  
3/ out = !(A && !(B)) ;    // out = ?
```

55  
chu\_yet\_hexes

## Phần 2 : Dữ liệu cơ sở và phép toán

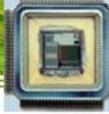


### Bài tập

```
1 <khai báo thư viện>  
2 void main()  
3 {  
4     .....  
5     .....  
6     .....  
7     .....  
8     .....  
9     .....  
10    .....  
11    .....  
12    .....  
13    .....  
14    .....  
15    .....
```

57  
chu\_yet\_hexes

## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

★ Tính toán: Toán tử so sánh

```
char A = 11;  char B = 0;
```

Cho biết kết quả của phép gán :

```
1/ out = A > B ;           // out = ?  
2/ out = A > ~ (B) ;       // out = ?  
3/ out = (A == B) ;         // out = ?  
4/ out = (A == 0xB) ;       // out = ?
```

59  
chu\_yet\_hexes

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

★ Viết chương trình nhập vào năm dương lịch, kiểm tra xem có phải năm nhuận.

Năm nhuận là năm chia hết cho 4 và không chia hết cho 100 hoặc là năm chia hết cho 400

56  
chu\_yet\_hexes

## Phần 2 : Dữ liệu cơ sở và phép toán



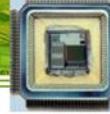
### Nhập, xuất, tính toán

★ Tính toán: Toán tử so sánh : true ( $\neq 0$ ), false ( $= 0$ )

Ký hiệu	Chức năng	# Toán hạng
>	Lớn hơn	2
<	Nhỏ hơn	2
>=	Lớn hơn hoặc bằng	2
<=	Nhỏ hơn hoặc bằng	2
==	Bằng	2
!=	Khác	2

58  
chu\_yet\_hexes

## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

★ Tính toán: Toán tử điều kiện

Ký hiệu	Chức năng	# Toán hạng
? :	Điều kiện	3

asssign Y=A ? B : C ;  
(nếu A đúng thì Y = B, ngược lại Y = C)

60  
chu\_yet\_hexes

## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

#### Tính toán: Toán tử điều kiện

`char A = 11; char B = 12;`

Cho biết kết quả của phép gán :

1/out = ( A > B ) ? A : B ; // out = ?

2/out = ((A - B)<0) ? A : B ;// out = ?

61  
chu\_0711\_2008

## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

#### Tính toán: Toán tử dịch

`char A = 11; char B = 15;`

Cho biết kết quả của phép gán :

1/out = A >> 2; // out = ?

2/out = B << 4; // out = ?

63  
chu\_0711\_2008

## Phần 2 : Dữ liệu cơ sở và phép toán



### Ký tự và bảng mã ASCII

#### Bảng mã ASCII

3210	000	001	010	011	100	101	110	111
0000	NUL	DLE	SPACE	0	Ø	P	~	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	:	K	[	k	{
1100	FF	FS	.	<	L	/	l	
1101	CR	GS	-	=	M	]	m	]
1110	SO	RS	-	>	N	-	n	-
1111	SI	US	/	?	O	-	o	DEL

65  
chu\_0711\_2008

## Phần 2 : Dữ liệu cơ sở và phép toán



### Nhập, xuất, tính toán

#### Tính toán: Toán tử dịch

Ký hiệu	Chức năng	# Toán hạng
>>	Dịch phải	2
<<	Dịch trái	2

62  
chu\_0711\_2008

## Phần 2 : Dữ liệu cơ sở và phép toán



### Ký tự và bảng mã ASCII

#### Ký tự

Kiểu char hoặc unsigned char

Một ký tự gồm 8 bit dựa trên bảng mã ASCII

64  
chu\_0711\_2008

## Phần 2 : Dữ liệu cơ sở và phép toán



### Ví dụ

#### Nhập vào 1 ký tự bất kỳ trên bàn phím và xuất ra mã ASCII của ký tự đó

66  
chu\_0711\_2008

## Phần 2 : Dữ liệu cơ sở và phép toán



### Ví dụ

```
1 <khai báo thư viện>
2 void main()
3 {
4     char KyTu;
5     printf ("Nhập 1 ký tự trên bàn phím:");
6     scanf ("%c",&KyTu);
7     printf ("Ký tự %c có mã ASCII là:%d",KyTu,
8 KyTu);
9     getch();
10 }
```

67  
chu\_yet\_jesus

## Phần 2 : Dữ liệu cơ sở và phép toán



### Ví dụ

#### ★ Kết quả

Nhập 1 ký tự trên bàn phím:  
Ký tự A có mã ASCII là:65

68  
chu\_yet\_jesus

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

★ Nhập vào 1 ký tự chữ thường bất kỳ trên bàn phím và xuất ra chữ hoa của ký tự đó.

Lưu ý: Nếu nhập sai (không phải chữ cái) sẽ không hiện ra kết quả

69  
chu\_yet\_jesus

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

★ Nhập vào 1 ký tự chữ hoa bất kỳ trên bàn phím và xuất ra chữ thường của ký tự đó.

Lưu ý: Nếu nhập sai (không phải chữ cái) sẽ không hiện ra kết quả

71  
chu\_yet\_jesus

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     char KyTu;
5     printf ("Nhập 1 ký tự chữ thường:");
6     .....
7
8     printf ("\nKý tự chữ hoa là:%c",KyTu);
9     getch();
10 }
```

70  
chu\_yet\_jesus

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     char KyTu;
5     printf ("Nhập 1 ký tự chữ hoa:");
6     .....
7
8     printf ("\nKý tự chữ thường là:%c",KyTu);
9     getch();
10 }
```

72  
chu\_yet\_jesus

## Phần 2 : Dữ liệu cơ sở và phép toán



### Phép gán tắt

★ $i++ / ++i$	$i = i + 1$
$i- / --i$	$i = i - 1$
$sum += n$	$sum = sum + n$
$N <= 2$	$N = N <= 2$
$n = m ++$	$n = m$
	$m = m + 1$
$n = ++m$	$m = m + 1$
	$n = m$

73

chu\_y\_moi\_moi

## Phần 2 : Dữ liệu cơ sở và phép toán



### Một số hàm toán học

★ $\sqrt{x}$	căn bậc 2 của x
$\text{pow}(x,y)$	$x$ lũy thừa $y$
$\text{exp}(x)$	$e$ lũy thừa $x$
$\log/\log_{10}(x)$	logarit cơ số $e$ / 10 của x
$\text{abs}(x)$	trị tuyệt đối của x
$\cos/\sin/\tan(x)$	$\cos$ / $\sin$ / $\tan$ của x
$\text{acos}/\text{asin}/\text{atan}(x)$	$\arccos$ / $\arcsin$ / $\arctan$ của x
$\text{floor}/\text{ceil}$	làm tròn xuống/lên của x

74

chu\_y\_moi\_moi

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

- ★ Nhập vào giá trị x. Tính các giá trị y. Với:

$$y_1 = \sqrt{x^3 + x^2}$$

$$y_2 = \sqrt[3]{x^5 - x^3}$$

75

chu\_y\_moi\_moi

## Phần 2 : Dữ liệu cơ sở và phép toán



### Bài tập

```

1 <khai báo thư viện>
2 void main()
3 {
4
5
6
7
8
9
10
11
12
13 }
```

76

chu\_y\_moi\_moi

## Phần 3 : Cấu trúc điều khiển



### Khối lệnh và biến

#### Khối lệnh (block)

Khối lệnh là tập hợp các lệnh nằm trong một cặp dấu ngoặc nhọn bắt đầu bằng { kết thúc bằng }

Các khối lệnh có thể lồng nhau, khối lệnh này nằm trong khối lệnh kia

Bên trong khối lệnh có thể khai báo biến dùng riêng cho khối lệnh đó gọi là biến cục bộ

77

chu\_y\_moi\_moi

## Phần 3 : Cấu trúc điều khiển



### Khối lệnh và biến

#### Biến có 2 loại

Biến cục bộ được khai báo trong một khối lệnh và chỉ có thể sử dụng trong khối lệnh đó và khối lệnh con (sub block) của nó

Biến toàn cục được khai báo nằm ngoài tất cả khối lệnh (thường được khai báo dưới khai báo thư viện)

#### Không thể đặt tên biến trùng nhau trong cùng khối lệnh

#### Hạn chế đặt tên biến trùng nhau giữa các khối lệnh lồng nhau

78

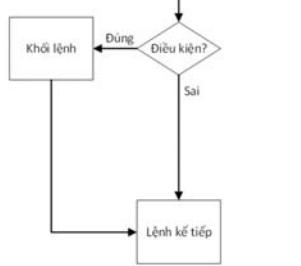
chu\_y\_moi\_moi

## Phần 3 : Cấu trúc điều khiển

### Cấu trúc rẽ nhánh

#### ★ Cấu trúc if else đơn giản

```
if (<điều kiện>)
{
    <khoi_lệnh>
}
```



Nếu điều kiện đúng thực hiện khối lệnh, nếu điều kiện sai thì bỏ qua

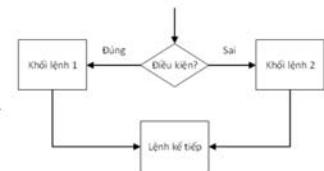
79  
chu\_yet\_hexes

## Phần 3 : Cấu trúc điều khiển

### Cấu trúc rẽ nhánh

#### ★ Cấu trúc if else đầy đủ

```
if (<điều kiện>)
{
    <khoi_lệnh 1>
}
else
{
    <khoi_lệnh 2>
}
```



Nếu điều kiện đúng thực hiện khối lệnh 1, nếu điều kiện sai thực hiện khối lệnh 2

80  
chu\_yet\_hexes

## Phần 3 : Cấu trúc điều khiển

### Bài tập

★ Nhập vào 2 giá trị a và b là số nguyên. So sánh và xuất ra kết quả a lớn hơn b, a nhỏ b hay a bằng b.

81  
chu\_yet\_hexes

## Phần 3 : Cấu trúc điều khiển

### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     int a, b;
5     printf ("Nhập a:");
6     scanf ("%d",&a);
7     printf ("Nhập b:");
8     scanf ("%d",&b);
9     if (a>b)
10         printf ("a=%d lon hon b=%d",a,b);
11     else
12     .....
13     .....
14     .....
15 }
```

82  
chu\_yet\_hexes

## Phần 3 : Cấu trúc điều khiển

### Bài tập

★ Nhập vào 1 ký tự bất kỳ trên bàn phím. Cho biết đó là chữ cái (thường hay hoa), chữ số, hay dấu. Xuất ra mã ASCII tương ứng của ký tự đó.

83  
chu\_yet\_hexes

## Phần 3 : Cấu trúc điều khiển

### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     char KyTu;
5     printf ("Nhập 1 ký tu bat ky:");
6     .....
7     .....
8     .....
9     .....
10    .....
11    .....
12    .....
13    .....
14    .....
15 }
```

84  
chu\_yet\_hexes

## Phần 3 : Cấu trúc điều khiển



### Cấu trúc rẽ nhánh

#### ★ Cấu trúc switch

```
switch (<biến>)
{
    case <giá trị 1>: <khoi lệnh 1>
        break;
    case <giá trị 2>: <khoi lệnh 2>
        break;
    case <giá trị n>: <khoi lệnh n>
        break;
    default: <khoi lệnh mặc định>
}
```

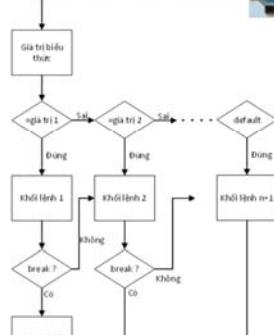
85  
chu\_trinh\_javac

## Phần 3 : Cấu trúc điều khiển



### Cấu trúc rẽ nhánh

#### ★ Cấu trúc switch

86  
chu\_trinh\_javac

## Phần 3 : Cấu trúc điều khiển



### Bài tập

★ Nhập vào 1 số từ 0 đến 10. Nếu là 0 xuất ra “khong”, 1 xuất ra “mot”, ... 10 xuất ra “muoi”. Nếu nằm ngoài giá trị từ 0 đến 10 thì báo “Nhap sai”.

87  
chu\_trinh\_javac

## Phần 3 : Cấu trúc điều khiển



### Bài tập

★ Nhập điểm nguyên từ 0 đến 10 cho sinh viên. Tính xếp loại cho sinh viên biết:

0 → 3: yếu

4: kém

5→6: trung bình

7: khá

8: giỏi

9→ 10: xuất sắc

88  
chu\_trinh\_javac

## Phần 3 : Cấu trúc điều khiển



### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     int So;
5     printf ("Nhap 1 so tu 0 den 10:");
6     scanf ("%d",&So);
7
8
9
10
11
12
13
14
15
```

88  
chu\_trinh\_javac

## Phần 3 : Cấu trúc điều khiển



### Bài tập

★ Nhập điểm nguyên từ 0 đến 10 cho sinh viên. Tính xếp loại cho sinh viên biết:

0 → 3: yếu

4: kém

5→6: trung bình

7: khá

8: giỏi

9→ 10: xuất sắc

89  
chu\_trinh\_javac

## Phần 3 : Cấu trúc điều khiển



### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     int Diem;
5     printf ("Nhap diem tu 0 den 10:");
6     scanf ("%d",&Diem);
7
8
9
10
11
12
13
14
15
```

90  
chu\_trinh\_javac

## Phần 3 : Cấu trúc điều khiển



### Cấu trúc lặp

- ★ Cấu trúc lặp dùng để lặp đi lặp lại khối lệnh nào đó phục vụ cho quá trình tính toán
- ★ Thông thường mỗi vòng lặp sẽ cập nhật lại giá trị các biến để đưa tới kết quả cuối cùng
- ★ Ngôn ngữ C có 3 cấu trúc lặp phổ biến: *while*, *do while*, *for*

91  
chu\_trinh\_jesus

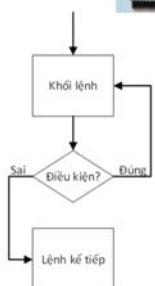
## Phần 3 : Cấu trúc điều khiển



### Cấu trúc lặp

#### ★ Cấu trúc *do while*

```
do
{
    <khối lệnh>
} while (<điều kiện>);
```



Cấu trúc *do while* sẽ thực hiện khối lệnh trước, kiểm tra điều kiện sau

93  
chu\_trinh\_jesus

## Phần 3 : Cấu trúc điều khiển



### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     char KyTu;
5     do
6     {
7         .....
8
9     } while (.....);
10    printf ("\nBan da nhan %c. \nThoat chuong
11 trinh", KyTu);
12    getch();
13 }
```

95  
chu\_trinh\_jesus

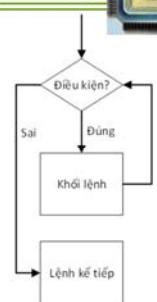
## Phần 3 : Cấu trúc điều khiển



### Cấu trúc lặp

#### ★ Cấu trúc *while*

```
while (<điều kiện>)
{
    <khối lệnh>
}
```



Nếu điều kiện đúng thì thực hiện khối lệnh liên tục cho đến khi điều kiện sai

92  
chu\_trinh\_jesus

## Phần 3 : Cấu trúc điều khiển



### Bài tập

★ Viết chương trình xuất liên tục các ký tự nhập từ bàn phím ra màn hình cho đến khi bấm 'q' hoặc 'Q' (quit).

★ Viết chương trình xuất liên tục các ký tự nhập từ bàn phím ra màn hình cho đến khi bấm phím ESC.

94  
chu\_trinh\_jesus

## Phần 3 : Cấu trúc điều khiển



### Bài tập

★ Viết chương trình xuất ra trên màn hình:

- Chuyen nganh May Tinh-He Thong Nhung
- Chuyen nganh Vien Thong-Mang
- Chuyen nganh Dien Tu

Bam [1-3] de chon chuyen nganh

Sau khi bấm phím [1-3] sẽ xuất hiện câu chào:

Ban da chon chuyen nganh <tên chuyên ngành>

Nếu bấm phím sai sẽ xuất hiện yêu cầu chọn lại

96  
chu\_trinh\_jesus

## Phần 3 : Cấu trúc điều khiển

### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     char KyTu;
5     .....
6     .....
7     .....
8     .....
9     .....
10    .....
11    .....
12    .....
13    .....
14    .....
15 }
```

97



## Phần 3 : Cấu trúc điều khiển

### Bài tập

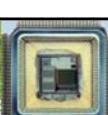
#### Kết quả

```
1. Chuyen nganh May Tinh-He Thong Nhung
2. Chuyen nganh Vien Thong-Mang
3. Chuyen nganh Dien Tu
Bam [1-3] de chon chuyen nganh:5
Nhap sai!!! Bam [1-3] de chon lai chuyen nganh:6
Nhap sai!!! Bam [1-3] de chon lai chuyen nganh:1
Ban da chon chuyen nganh May Tinh-He Thong Nhung.
```

98



## Phần 3 : Cấu trúc điều khiển



### Cấu trúc lặp

#### Cấu trúc for

```
for(int i=0, i<n, i++)
{
    <khối lệnh>
}
```

Cấu trúc for sẽ được sử dụng trong trường hợp biết trước số lần lặp. Trong đoạn code trên khối lệnh sẽ được thực hiện n lần.

99



## Phần 3 : Cấu trúc điều khiển

### Bài tập

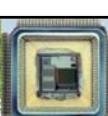
#### Viết chương trình nhập vào một số nguyên dương n bất kỳ. Tính biểu thức sau:

$$S_n = \frac{1}{1^2} + \frac{1}{2^2} + \dots + \frac{1}{n^2}$$

100



## Phần 3 : Cấu trúc điều khiển



### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     int n; double S,x; S=0;
5     printf ("Nhập số nguyên dương:");
6     scanf ("%d", &n);
7     for (.....)
8     {
9         .....
10        .....
11        .....
12        .....
13        .....
14        .....
15 }
```

101



## Phần 3 : Cấu trúc điều khiển

### Bài tập

#### Viết chương trình nhập vào một số nguyên dương n bất kỳ. Tính $S = n!$ ( $n$ giai thừa)

102



### Phần 3 : Cấu trúc điều khiển



#### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     int n; long S; S=0;
5     printf ("Nhập số nguyên dương:");
6     scanf ("%d",&n);
7     for (.....)
8     {
9         .....
10        .....
11        .....
12        .....
13        .....
14        .....
15 }
```

103  
ctrc\_jrml\_jmav

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ **Viết chương trình nhập vào một số nguyên dương n bất kỳ. Tính  $S = 1! + 2! + 3! + \dots + n!$**

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     int n; long S; S=0;
5     printf ("Nhập số nguyên dương:");
6     scanf ("%d",&n);
7     for (.....)
8     {
9         .....
10        .....
11        .....
12        .....
13        .....
14        .....
15 }
```

105  
ctrc\_jrml\_jmav

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ **Viết chương trình nhập vào một số nguyên dương n bất kỳ. Tính biểu thức sau:**

$$S_n = 1 + \frac{1}{1+2} + \frac{1}{1+2+3} + \dots + \frac{1}{1+2+3\dots+n}$$

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     .....
5     .....
6     .....
7     .....
8     .....
9     .....
10    .....
11    .....
12    .....
13    .....
14    .....
15 }
```

107  
ctrc\_jrml\_jmav

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ **Viết chương trình nhập vào một số nguyên dương N. Xuất dãy Fibonacci theo giá trị n nhập vào. Biết rằng dãy (n=8) sẽ bắt đầu như sau:**

0    1    1    2    3    5    8    13    21

(giá trị tiếp theo sẽ bằng tổng 2 giá trị trước)

108  
ctrc\_jrml\_jmav

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     int n;
5     printf ("Nhập số nguyên dương:");
6     scanf ("%d", &n);
7
8
9
10
11
12
13
14
15
```

109  
chu\_truc\_dieu\_kien

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ Viết chương trình nhập vào một số nguyên dương N. Kiểm tra N có phải là số nguyên tố.
- ★ Viết chương trình nhập vào một số nguyên dương N. Xuất ra tất cả số nguyên tố nhỏ hơn hoặc bằng N.
- ★ Viết chương trình tìm tất cả số nguyên dương có 3 chữ số abc sao cho

$$abc = a^3 + b^3 + c^3$$

Ví dụ:  $153 = 1^3 + 5^3 + 3^3$

111  
chu\_truc\_dieu\_kien

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ Viết chương trình nhập vào một số nguyên dương bất kỳ. Xuất ra giá trị nhị phân của số đó.
- ★ Viết chương trình nhập điểm 4 môn A, B, C, D. Chuyên ngành MT-HTN xét điểm môn A, B. Chuyên ngành VT-M xét điểm môn B, C. Chuyên ngành ĐT xét điểm môn A, D. Biết rằng quy định tổng điểm 2 môn xét chuyên ngành phải lớn hơn hoặc bằng 10 mới được vào chuyên ngành tương ứng. Xuất ra kết quả xét vào chuyên ngành theo mức độ ưu tiên để tối ưu sự lựa chọn.

113  
chu\_truc\_dieu\_kien

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ Viết chương trình nhập vào một số nguyên dương n bất kỳ. Tính biểu thức sau:

$$S_n = \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

- ★ Viết chương trình giải phương trình bậc nhất  $ax+b=0$

- ★ Viết chương trình giải phương trình bậc hai  $ax^2+bx+c=0$

110  
chu\_truc\_dieu\_kien

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ Viết chương trình in ra bảng cửu chương với vòng lặp for lồng nhau.
- ★ Viết chương trình nhập vào một số nguyên dương N. In ra chữ số lớn nhất và chữ số nhỏ nhất của nó.  
Ví dụ: N=429      chữ số lớn nhất là 9      nhỏ nhất là 2
- ★ Viết chương trình tìm ước số chung lớn nhất, bộ số chung nhỏ nhất của 2 số nguyên dương nhập vào.

112  
chu\_truc\_dieu\_kien

### Phần 3 : Cấu trúc điều khiển



#### Cấu trúc lặp

- ★ Chỉ thị break và continue  
Chỉ thị break dùng để thoát khỏi vòng lặp ngay lập tức.  
Chỉ thị continue dùng để chuyển qua vòng lặp mới ngay lập tức

114  
chu\_truc\_dieu\_kien

### Phần 3 : Cấu trúc điều khiển



#### Trình bày màn hình console

- ★ Đặt con trỏ màn hình vào tọa độ x, y với x là cột (80 cột), y là dòng (25 dòng)

```
gotoxy(x,y);
```

- ★ Xóa màn hình

```
clrscr();
```

115  
ctrl\_ruru\_jesus

### Phần 3 : Cấu trúc điều khiển



#### Trình bày màn hình console

- ★ Các màu trong C

BLACK (0)	LIGHTGRAY (7)	YELLOW (14)
BLUE (1)	DARKGRAY (8)	WHITE (15)
GREEN (2)	LIGHTBLUE (9)	BLINK (128)
CYAN (3)	LIGHTGREEN (10)	
RED (4)	LIGHTCYAN (11)	
MAGENTA (5)	LIGHTRED (12)	
BROWN (6)	LIGHTMAGENTA (13)	

117  
ctrl\_ruru\_jesus

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

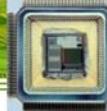
- ★ Viết chương trình xuất ra các dòng sau với nền vàng chữ đỏ có nhấp nháy ngay giữa màn hình console

KHOA DIEN TU – VIEN THONG

CHUYEN NGANH MAY TINH-HE THONG NHUNG

119  
ctrl\_ruru\_jesus

### Phần 3 : Cấu trúc điều khiển



#### Trình bày màn hình console

- ★ Màu chữ (16 màu)  
`textcolor(<màu>);`

- ★ Màu nền (8 màu)  
`textbackground(<màu>);`

118  
ctrl\_ruru\_jesus

### Phần 3 : Cấu trúc điều khiển



#### Trình bày màn hình console

- ★ Xác lập thuộc tính ký tự với 8 bit  
`textattr(<thuộc tính 8bit>);`

B	b	b	b	f	f	f	f	f
---	---	---	---	---	---	---	---	---

B: Blink\_nhấp nháy

b: Background\_màu nền

f: foreground\_màu chữ

118  
ctrl\_ruru\_jesus

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ Viết chương trình hiện ra trên màn hình với n là số hàng nhập từ bàn phím. Ví dụ: n=5

```
*  
**  
***  
****  
*****
```

120  
ctrl\_ruru\_jesus

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ Viết chương trình hiện ra trên màn hình với n là số hàng nhập từ bàn phím. Ví dụ: n=5

```
*  
***  
*****  
*****  
*****
```

121  
ctrl\_ruru\_hexes

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ Viết chương trình hiện ra trên màn hình với n là số hàng nhập từ bàn phím. Ví dụ: n=5

```
1  
232  
34543  
4567654  
567898765
```

122  
ctrl\_ruru\_hexes

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ Viết chương trình cho chữ A chạy liên tục theo chiều kim đồng hồ xung quanh 4 cạnh màn hình console cho đến khi bấm phím bất kỳ để thoát

```
kbhlt(); //kiểm tra ký tự bất kỳ gõ từ bàn phím  
delay(<đơn vị ms>); //tri hoán một đơn vị thời  
//gian tĩnh bằng mili giây  
//thư viện dos.h
```

123  
ctrl\_ruru\_hexes

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ Viết chương trình tạo hiệu ứng như sau:  
Dòng đầu tiên dịch từ trái sang phải  
Sau đó, dòng thứ hai dịch từ phải sang trái  
Khi dịch màu chữ đổi liên tục

KHOA DIEN TU – VIEN THONG

CHUYEN NGANH MAY TINH-HE THONG NHUNG

124  
ctrl\_ruru\_hexes

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ Viết chương trình trò chơi tung xí ngầu (xúc xác) với cách chơi như sau:

B1: Chọn số người chơi



B2: Chọn số lần tung

B3: Mỗi người lần lượt tung, báo điểm sau mỗi lượt

B4: Người nào có số điểm lớn nhất sẽ thắng

Tạo giao diện trò chơi với màu sắc và hình ảnh nổi bật

125  
ctrl\_ruru\_hexes

### Phần 3 : Cấu trúc điều khiển



#### Bài tập

- ★ Viết chương trình trò chơi “Chiếc nón kỳ diệu” (quay nón đoán ô chữ)

B1: Người quản trò chọn câu cần đoán

B2: Người chơi quay nón (tung xí ngầu) để chọn 1 trong các trường hợp: mắt lượn, nhân dôi điếm, chia dôi điếm, chọn chữ tương ứng mỗi chữ 100 điểm, lật chữ tại ô bất kỳ, đoán câu...

B3: Người có số điểm cao nhất sẽ thắng

126  
ctrl\_ruru\_hexes

## Phần 4 : Chương trình con



### Chương trình con

- ★ Chương trình lớn phải được phân chia thành các phần nhỏ để dễ dàng quản lý, phân công, bảo trì, mở rộng và tái sử dụng.
- ★ Chương trình con có nhiệm vụ cụ thể và độc lập
- ★ Trong ngôn ngữ C chỉ có duy nhất một loại chương trình con là hàm (function), thủ tục (procedure) được xem là một dạng hàm đặc biệt.

127  
chu\_trinh\_con

## Phần 4 : Chương trình con

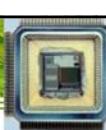


### Chương trình con

- ★ Hàm `main()` là một dạng chương trình con đặc biệt.  
Hàm `main()` là duy nhất trong một chương trình.
- ★ Một số hàm khác đều có sẵn trong các thư viện.
- ★ Nếu một hàm trả về giá trị `void` (rỗng) thì hàm đó là một dạng của thủ tục.

128  
chu\_trinh\_con

## Phần 4 : Chương trình con



### Chương trình con

- ★ Khai báo hàm
  - <kiểu dữ liệu trả về> <tên hàm> <danh sách tham số>
- ★ Chỉ thị `return` để trả về kết quả
- ★ Nếu hàm có <kiểu dữ liệu trả về> là `void` thì không cần chỉ thị `return`

129  
chu\_trinh\_con

## Phần 4 : Chương trình con



### Bài tập

- ★ Viết hàm tìm ước số chung lớn nhất, bội số chung nhỏ nhất của 2 số nguyên dương. Sau đó viết chương trình sử dụng hàm trên.

130  
chu\_trinh\_con

## Phần 4 : Chương trình con



### Bài tập

```
1 <khai báo thư viện>
2 int USCLN(int a, int b)
3 {
4     while ((a!=0) && (b!=0))
5         if (a > b) a = a % b;
6         else b = b % a;
7
8     if (a==0) return b;
9     else return a;
10 }
11 int BSCNN(int a, int b)
12 {
13     return (a * b)/USCLN(a,b);
14 }
```

131  
chu\_trinh\_con

## Phần 4 : Chương trình con



### Bài tập

```
16 void main ()
17 {
18     int n, m;
19     printf ("Nhập n=");
20     scanf ("%d",&n);
21     printf ("Nhập m=");
22     scanf ("%d",&m);
23
24
25
26
27
28     getch();
29 }
30
```

132  
chu\_trinh\_con

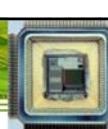


### Bài tập

- ★ Nhập vào 1 số nguyên dương N có tối đa 4 chữ số. Xuất ra cách đọc của số đó.

Ví dụ: N=4321

Cách đọc: Bon ngan ba tram hai muoi mot



### Tham số

- ★ Tham số là danh sách các biến ngay sau tên hàm để khi gọi hàm thì các tham số này trở thành biến thực sự
- ★ Khi gọi hàm, việc truyền các biến đến các tham số gọi là truyền tham số
- ★ Có 2 cách truyền tham số: truyền giá trị và truyền địa chỉ



### Tham số

#### Ví dụ

```
void tang2 (int a)
{
    a=a+2;
}

void main ()
{
    int a = 4;
    tang2 (a);
    printf("Gia tri cua a=%d",a); // a= ???
    getch();
}
```



### Bài tập

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....



### Tham số

- ★ Truyền giá trị tham số nghĩa là giá trị lưu trong biến được truyền đến tham số
- ★ Mọi sự thay đổi giá trị của biến trong hàm sẽ không làm thay đổi giá trị của biến được truyền đến



### Tham số

- ★ Truyền địa chỉ của biến được truyền đến tham số
- ★ Mọi sự thay đổi giá trị của biến trong hàm sẽ làm thay đổi giá trị của biến được truyền đến
- ★ Khi khai báo hàm dùng dấu \* trước tham số, khi gọi hàm dùng dấu & trước biến

## Phần 4 : Chương trình con



### Tham số

#### Ví dụ

```
void tang2 (int *a)      //dấu * trước tham số
{
    *a=*a+2;
}

void main ()
{
    int a = 4;
    tang2 (&a);          // dấu & trước tên biến
    printf("Gia tri cua a=%d",a); // a= ???
    getch();
}
```

139  
chu\_2013\_javas

## Phần 4 : Chương trình con



### Tham số

#### Truyền tham chiếu tham số của C++

140  
chu\_2013\_javas

## Phần 4 : Chương trình con



### Tham số

#### Ví dụ

```
void tang2 (int &a)      //dấu & trước tham số
{
    a=a+2;
}

void main ()
{
    int a = 4;
    tang2 (a);
    printf("Gia tri cua a=%d",a); // a= ???
    getch();
}
```

141  
chu\_2013\_javas

## Phần 4 : Chương trình con



### Tham số

#### Xem xét 2 ví dụ tương đương khi viết hàm Sum tính tổng 2 số

142  
chu\_2013\_javas

## Phần 4 : Chương trình con



### Tham số

#### Ví dụ

```
int Sum (int a, int b)
{
    return a + b;
}

void main ()
{
    int a = 4, b = 8; int Tong;
    Tong = Sum (a , b);
    printf("Tong cua a va b la %d",Tong);
    getch();
}
```

143  
chu\_2013\_javas

## Phần 4 : Chương trình con



### Tham số

#### Ví dụ

```
void Sum (int &Tong, int a, int b)
{
    Tong = a + b;
}

void main ()
{
    int a = 4, b = 8; int Tong;
    Sum (Tong , a , b);
    printf("Tong cua a va b la %d",Tong);
    getch();
}
```

144  
chu\_2013\_javas

## Phần 4 : Chương trình con



### Bài tập

- ★ Viết hàm tính tiền cước taxi biết rằng:

Tham số đầu vào: số km đi, điều kiện đường đi

Điều kiện 1:

Km đầu: 20.000đ

Km 2 đến 10: 15.000đ

Lớn hơn 10: 10.000đ

Điều kiện 2:

Km đầu: 30.000đ

Km 2 đến 5: 25.000đ

Lớn hơn 5: 20.000đ

145  
chu\_trinh\_java

## Phần 4 : Chương trình con



### Bài tập

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

146  
chu\_trinh\_java

## Phần 5 : Dữ liệu mảng



### Tổng quan

- ★ Mảng là kiểu dữ liệu có cấu trúc dùng để chứa tập các phần tử có cùng kiểu dữ liệu
- ★ Kiểu mảng được hỗ trợ dưới dạng mảng một chiều hoặc nhiều chiều trong ngôn ngữ C
- ★ Trong thực tế mảng một chiều được sử dụng nhiều nhất

147  
chu\_trinh\_java

## Phần 5 : Dữ liệu mảng



### Khai báo

- ★ Với một mảng có n phần tử, phần tử đầu tiên là 0, phần tử cuối cùng là n-1
- ★ Vòng lặp for thường xuyên được sử dụng khi làm việc với mảng
- ★ Có thể khởi tạo giá trị ban đầu cho mảng

```
int DaySoNguyen[10]={9,5,6,4,5,1,2,8,3,9};
```

149  
chu\_trinh\_java

## Phần 5 : Dữ liệu mảng



### Khai báo

- ★ Mảng một chiều đơn giản gồm các phần tử có cùng kiểu dữ liệu xếp thành dãy

#### Khai báo

<kiểu dữ liệu> <tên mảng> <số phần tử>

#### Ví dụ :

```
int DaySoNguyen[10];
```

Khai báo dãy số nguyên có tên DaySoNguyen kiểu integer với 10 phần tử

148  
chu\_trinh\_java

## Phần 5 : Dữ liệu mảng



### Khai báo

- ★ Nếu không liệt kê đầy đủ giá trị khởi tạo. Ví dụ:  

```
int DaySoNguyen[10]={9,5,6};
```

Những phần tử đầu tiên sẽ nhận giá trị 9, 5, 6. Các phần tử còn lại nhận giá trị 0.

#### Truyền mảng 1 chiều cho hàm

```
void Mang1Chieu (int a[]) {<khoi lệnh>} //viết hàm  
Mang1Chieu (a); //gọi hàm
```

150  
chu\_trinh\_java

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình nhập mảng 1 chiều gồm 10 số nguyên. Sau đó xuất ra mảng đó theo thứ tự ngược lại.

151  
chu\_2013\_javas

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình nhập 2 mảng 1 chiều A và B gồm 10 số nguyên. Sau đó xuất ra mảng S với mỗi phần tử của mảng S là tổng của mỗi phần tử mảng A và mảng B.

153  
chu\_2013\_javas

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình nhập mảng 1 chiều gồm 10 số nguyên. Tim phần tử số nguyên âm lớn nhất và vị trí của nó.

Ví dụ: có mảng các số nguyên

-8    3    -3    -5    7    9    4    6

-3 là số nguyên âm lớn nhất

155  
chu\_2013\_javas

## Phần 5 : Dữ liệu mảng



### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     int DaySoNguyen[10];
5     printf ("Nhập dãy số nguyên 10 phần tử:");
6     for (int i=0;i<10;i++)
7     {
8         scanf ("%d",&DaySoNguyen[i]);
9     }
10    printf ("\nXuất dãy số nguyên ngược lại:");
11    for (int i=0;i<10;i++)
12    {
13        .....
14    }
15 }
```

152  
chu\_2013\_javas

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình nhập mảng 1 chiều gồm 10 số nguyên. Tim phần tử lớn nhất, nhỏ nhất và vị trí của nó.

154  
chu\_2013\_javas

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình nhập mảng 1 chiều gồm 10 số nguyên. Tim phần tử số nguyên âm lớn nhất và vị trí của nó.

Ví dụ: có mảng các số nguyên

-8    3    -3    -5    7    9    4    6

-3 là số nguyên âm lớn nhất

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình nhập mảng 1 chiều gồm 10 số nguyên. Sau đó nhập vào 1 giá trị. Tim xem giá trị đó có trong mảng hay không? Ở vị trí nào?

156  
chu\_2013\_javas

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình tạo mảng 1 chiều ngẫu nhiên gồm 100 số nguyên có giá trị từ 0 → 30. Tìm xem trong mảng những số nào trùng nhau? Mỗi số trùng nhau mấy lần?

157  
chu\_trinh\_javas

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình tạo mảng 1 chiều ngẫu nhiên gồm 10 số nguyên. Sau đó nhập vào 1 giá trị. Chèn giá trị này vào vị trí đầu tiên của mảng. Giá trị phần tử cuối cùng sẽ mất đi.

159  
chu\_trinh\_javas

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình tạo mảng 1 chiều ngẫu nhiên gồm 10 số nguyên. Sau đó cho chọn lựa xóa hay chèn.  
Nếu xóa thì nhập vị trí cần xóa.  
Nếu chèn thì nhập vị trí kèm giá trị cần chèn vào mảng.

161  
chu\_trinh\_javas

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình tạo 2 mảng A và B ngẫu nhiên gồm 100 số nguyên có giá trị từ 0 → 99. Tìm xem số nào có trong A mà không có trong B? Tìm xem số nào xuất hiện trong cả 2 mảng A và B?

158  
chu\_trinh\_javas

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình tạo mảng 1 chiều ngẫu nhiên gồm 10 số nguyên. Sau đó nhập vào 1 vị trí. Xóa bỏ phần tử tại vị trí này. Các phần tử còn lại sẽ dồn lên. Giá trị phần tử cuối cùng sẽ bằng zero.

160  
chu\_trinh\_javas

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình nhập mảng 1 chiều gồm 10 số nguyên. Trong quá trình nhập nếu giá trị bị trùng với giá trị trước thì yêu cầu nhập lại.

162  
chu\_trinh\_javas

## Phần 5 : Dữ liệu mảng



### Sắp xếp (sort)

★ Có nhiều thuật toán khác nhau

- Insertion sort
- Selection sort
- Bubble sort
- Heap sort
- Quick sort
- Merge sort
- ....

163  
chuoi\_nganh

## Phần 5 : Dữ liệu mảng



### Chuỗi

★ Chuỗi là mảng 1 chiều các ký tự hay còn gọi là mảng ký tự

### Ví dụ

```
char TenSV[80];  
char ChuyenNganh[40];
```

165  
chuoi\_nganh

## Phần 5 : Dữ liệu mảng



### Chuỗi

★ Một số lệnh dùng cho chuỗi (string.h)

```
strlen(<tên chuỗi>); // trả về số ký tự của chuỗi  
strcat(<chuỗi 1>,<chuỗi 2>); // ghép chuỗi  
strcpy(<chuỗi 1>,<chuỗi 2>); // sao chép chuỗi  
strcmp(<chuỗi 1>,<chuỗi 2>); // so sánh chuỗi  
// trả về 0 nếu chuỗi 1 = chuỗi 2
```

167  
chuoi\_nganh

## Phần 5 : Dữ liệu mảng



### Bài tập

★ Viết chương trình tạo mảng 1 chiều ngẫu nhiên gồm 100 số nguyên. Sau đó xuất ra mảng đó theo thứ tự từ nhỏ đến lớn.

164  
chuoi\_nganh

## Phần 5 : Dữ liệu mảng



### Chuỗi

★ Chuỗi kết thúc bởi ký tự NULL (ký tự đầu tiên trong bảng mã ASCII), ký hiệu là ‘\0’

★ Khởi tạo chuỗi có thể không cần khai báo số ký tự  
char ChuyenNganh[]="May tinh-He thong nhung";

Chương trình tự hiểu chuỗi trên có 24 ký tự (bao gồm 23 ký tự hiển thị và 1 ký tự NULL kết thúc chuỗi)

166  
chuoi\_nganh

## Phần 5 : Dữ liệu mảng



### Chuỗi

★ Một số lệnh dùng cho chuỗi (string.h)

```
strlen(<tên chuỗi>); // trả về số ký tự của chuỗi  
strcat(<chuỗi 1>,<chuỗi 2>); // ghép chuỗi  
strcpy(<chuỗi 1>,<chuỗi 2>); // sao chép chuỗi  
strcmp(<chuỗi 1>,<chuỗi 2>); // so sánh chuỗi  
// trả về 0 nếu chuỗi 1 = chuỗi 2
```

## Phần 5 : Dữ liệu mảng



### Bài tập

★ Viết chương trình nhập chuỗi bất kỳ. Cho biết tổng số ký tự của chuỗi.

168  
chuoi\_nganh

## Phần 5 : Dữ liệu mảng

### Bài tập

```
1 <khai báo thư viện>
2 void main()
3 {
4     char ChuoiNhap[40];
5     printf ("Nhập chuỗi bất kỳ:");
6
7
8
9     getch();
10 }
```

169  
chuoi\_bat\_ky.c

## Phần 5 : Dữ liệu mảng

### Bài tập

#### Kết quả

Nhập chuỗi bất kỳ: May Tinh-He Thong Nhung  
Chuỗi nhập là "May Tinh-He Thong Nhung" có số ký tự là 23.

170  
chuoi\_bat\_ky.c

## Phần 5 : Dữ liệu mảng

### Bài tập

- ★ Viết chương trình nhập chuỗi bất kỳ. Chuyển chuỗi thành chữ hoa (không dùng thư viện string.h)
- ★ Viết chương trình nhập chuỗi bất kỳ. Đếm xem trong chuỗi có bao nhiêu chữ 'A'.
- ★ Viết chương trình nhập chuỗi bất kỳ. Liệt kê trong chuỗi có bao nhiêu chữ từ 'A' đến 'Z'.  
**Lưu ý:** sử dụng thêm 1 mảng để lưu trữ số lần xuất hiện của mỗi chữ cái từ 'A' đến 'Z'.

171  
chuoi\_bat\_ky.c

## Phần 5 : Dữ liệu mảng

### Con trỏ (pointer)

- ★ Con trỏ là biến chứa địa chỉ, không chứa dữ liệu như các biến thông thường

#### Khai báo

```
int *p; //khai báo con trỏ p
int x=35;
p=&x; //p chứa địa chỉ của biến x
```

173  
chuoi\_bat\_ky.c

## Phần 5 : Dữ liệu mảng

### Bài tập

- ★ Viết chương trình nhập họ và tên. Chuyển thành 2 chuỗi họ và tên riêng biệt.  
**Ví dụ:** "Nguyen Van An" thành "Nguyen Van" và "An"
- ★ Viết chương trình nhập họ và tên theo chữ thường. Chuyển các ký tự đầu thành chữ hoa  
**Ví dụ:** "nguyen van an" thành "Nguyen Van An"
- ★ Viết chương trình nhập họ và tên. Chuyển tên lên trước, họ ở sau như trong tiếng Anh  
**Ví dụ:** "Nguyen Van An" thành "An Van Nguyen"

172  
chuoi\_bat\_ky.c

## Phần 5 : Dữ liệu mảng

### Con trỏ (pointer)

- ★ Tên mảng (chuỗi) chính là con trỏ chỉ vào phần tử đầu tiên trong mảng

#### Ví dụ

```
char ChuyenNganh[]="May tinh-He thong nhung";
ChuyenNganh chính là con trỏ chứa địa chỉ phần tử đầu tiên trong chuỗi. Như vậy:
ChuyenNganh tương đương &ChuyenNganh[0]
*ChuyenNganh tương đương ChuyenNganh[0]
```

174  
chuoi\_bat\_ky.c

## Phần 5 : Dữ liệu mảng

### Ví dụ

```
1 <khai báo thư viện>
2 void main()
3 {
4     char ChuyenNganh[]="May tinh-He thong nhung";
5     printf ("%p tuong duong %p",ChuyenNganh,&ChuyenNganh[0]);
6     printf ("\n%c tuong duong %c",*ChuyenNganh,ChuyenNganh[0]);
7     getch();
8 }
9 //in giá trị con trỏ hay địa chỉ sử dụng %p
```

175  
chu\_tern\_hexes

## Phần 5 : Dữ liệu mảng

### Mảng 2 chiều

- ★ Mảng 2 chiều (hoặc hơn) thường được xử lý như mảng 1 chiều trong hầu hết các bài toán thực tế. Ví dụ: xử lý ảnh, video, tính toán ma trận...
- ★ Khi gặp mảng nhiều chiều nên xem như nó là mảng 1 chiều để xử lý

177  
chu\_tern\_hexes

## Phần 5 : Dữ liệu mảng

### Mảng 2 chiều

- ★ Có thể sử dụng con trỏ để truy xuất mảng nhiều chiều như mảng 1 chiều.
- ★ Ví dụ xuất ra các phần tử trong ma trận 2 x 2 theo cách thông thường

```
int mt[2,2] = { {1,2} , {3,4} }; int i,j;
for(i=0;i<2;i++)
{
    for(j=0;j<2;j++) printf("%d\t",mt[i,j]);
    printf ("\n");
}
```

179  
chu\_tern\_hexes

## Phần 5 : Dữ liệu mảng

### Con trỏ (pointer)

- ★ Có thể sử dụng con trỏ thay thế cách viết thông thường khi truy cập mảng/ chuỗi bất kỳ

- ★ Ví dụ in các phần tử trong dãy số nguyên 10 phần tử
  - int n[10]={9,5,6,4,5,1,2,8,3,9}; int i;
  - for(i=0;i<10;i++) printf("\n%d",n[i]);
  - hoặc
  - for(i=0;i<10;i++) printf("\n%d",\*(n+i));

176  
chu\_tern\_hexes

## Phần 5 : Dữ liệu mảng

### Mảng 2 chiều

- ★ Khai báo ma trận vuông 2 x 2 số nguyên
  - int matrix [2,2] = { {1,2} , {3,4} };
- ★ Về bản chất bộ nhớ máy tính không thể lưu trữ mảng nhiều chiều. Nó chỉ có thể lưu trữ theo 1 chiều. Do đó với ma trận n x m nó sẽ lưu trữ hàng 0 trước, sau đó tới hàng 1, hàng 2 ..., hàng n. Do đó
  - int matrix [2,2] = { {1,2} , {3,4} };
  - tương đương
  - int matrix [2,2] = { 1 , 2 , 3 , 4 };

178  
chu\_tern\_hexes

## Phần 5 : Dữ liệu mảng

### Mảng 2 chiều

- ★ Ví dụ xuất ra các phần tử trong ma trận 2 x 2 theo con trỏ
  - for(i=0;i<2;i++)
{
 for(j=0;j<2;j++) printf("%d\t",\*(mt\*i+j));
 printf ("\n");
}

180  
chu\_tern\_hexes

## Phần 5 : Dữ liệu mảng



### Bài tập

- ★ Viết chương trình tính tổng 2 ma trận n x m với n và m nhập từ bàn phím
- ★ Viết chương trình nhân 2 ma trận vuông n x n với n nhập từ bàn phím

Có thể sử dụng hàm rand() trong stdlib.h để tạo giá trị ngẫu nhiên

181  
chu\_2011\_javac

## Phần 6 : Dữ liệu cấu trúc



### Tổng quan

- ★ Một dữ liệu trong thực tế gồm nhiều thành phần ghép lại. Ví dụ: 1 sinh viên bao gồm họ và tên, mã số sinh viên, khoa ...

★ Khai báo kiểu dữ liệu cơ sở cho mỗi sinh viên  
char MSSV[8];  
char HoTenSV[30];  
char Khoa[40];

.....

182  
chu\_2011\_javac

## Phần 6 : Dữ liệu cấu trúc



### Khai báo

- ★ Kiểu dữ liệu cấu trúc cho phép ghép các kiểu dữ liệu cơ sở lại thành một kiểu dữ liệu duy nhất để tiện khai báo, quản lý

### Cú pháp

```
struct <tên kiểu cấu trúc>
{
    <kiểu dữ liệu cơ sở> <tên thành phần 1>
    <kiểu dữ liệu cơ sở> <tên thành phần 2>
    ...
};
```

183  
chu\_2011\_javac

## Phần 6 : Dữ liệu cấu trúc



### Khai báo

★ Ví dụ  
struct SinhVien  
{  
 char MSSV[8];  
 char HoTen[30];  
 char Khoa[40];  
 int Lop;  
} SV1, SV2;

184  
chu\_2011\_javac

## Phần 6 : Dữ liệu cấu trúc



### Khai báo

- ★ Ta có thể sử dụng `typedef` để định nghĩa kiểu dữ liệu mới

```
typedef struct SinhVien
{
    char MSSV[8];
    char HoTen[30];
    char Khoa[40];
    int Lop;
};

SinhVien SV1, SV2;
```

185  
chu\_2011\_javac

## Phần 6 : Dữ liệu cấu trúc



### Truy xuất

★ Để truy xuất đến thành phần trong cấu trúc có thể dùng dấu '.'  
strcpy(SV1.MSSV, "1820001");  
strcpy(SV1.HoTen, "Tran Van A");  
strcpy(SV1.Khoa, "Dien Tu- Vien Thong");  
SV1.Lop=1;

186  
chu\_2011\_javac

## Phần 6 : Dữ liệu cấu trúc

### Ví dụ

```
1 <khai báo thư viện>
2 void main()
3 {
4     typedef struct SinhVien
5     {
6         char MSSV[8];
7         char HoTen[30];
8         char Khoa[40];
9         int Lop;           //1 hoặc 2
10    };
11    SinhVien SV1, SV2;
12    strcpy(SV1.MSSV, "1820001");
13    strcpy(SV1.HoTen, "Tran Van A");
14    strcpy(SV1.Khoa, "Dien Tu - Vien Thong");
15    SV1.Lop=1;
16    printf("MSSV:%s\nHo Ten:%s",SV1.MSSV,SV1.HoTen);
17    printf("\nKhoa:%s\nLop:%d",SV1.Khoa,SV1.Lop);
18    getch();
19 }
```

187

## Phần 6 : Dữ liệu cấu trúc

### Ví dụ

#### Kết quả

```
MSSV:1820001
Ho Ten:Tran Van A
Khoa:Dien Tu - Vien Thong
Lop:1
```

188

## Phần 6 : Dữ liệu cấu trúc

### Bài tập

- ★ Viết chương trình tạo ra struct thời gian (gồm giờ, phút, giây). Nhập dữ liệu vào 2 biến kiểu struct thời gian và tính khoảng cách giữa 2 mốc thời gian.
- ★ Viết chương trình tạo ra struct ngày tháng (gồm ngày, tháng, năm). Nhập dữ liệu vào 2 biến kiểu struct ngày tháng và tính khoảng cách giữa 2 mốc ngày tháng.

189

## Phần 6 : Dữ liệu cấu trúc

### Bài tập

- ★ Viết chương trình tạo ra struct sinh viên gồm: mã số, họ tên, lớp, điểm trung bình.
  - Tạo ra danh sách 10 sinh viên và thực hiện tùy chọn:
  - Xuất danh sách ra màn hình
  - Tim sinh viên có điểm trung bình cao nhất, thấp nhất
  - Xếp danh sách theo mã số tăng dần
  - Xếp danh sách theo điểm trung bình giảm dần
  - Thêm/xóa sinh viên

190

## Phần 6 : Dữ liệu cấu trúc

### Bài tập

- ★ Viết chương trình tạo ra struct lớp học gồm: tên lớp, số lượng sinh viên (mảng struct sinh viên)
  - Tạo ra danh sách 3 lớp và thực hiện tùy chọn:
    - Tim sinh viên có điểm trung bình cao nhất, thấp nhất trong 3 lớp
    - Tim lớp giỏi/kém nhất dựa vào điểm trung bình
    - Xuất danh sách theo điểm trung bình giảm dần cho sinh viên cả 3 lớp

191

## Phần 7 : Tập tin

### Tổng quan

- ★ Tập tin (file) gồm một tập hợp các thông tin được lưu trữ trên máy tính
  - Tập tin sẽ được lưu trữ vào bộ nhớ ngoài nên không bị ảnh hưởng khi mất điện (do tắt máy), có thể truy xuất lại bất kỳ lúc nào

192

## Phần 7 : Tập tin



### Tổng quan

- ★ Tập tin là kiểu dữ liệu có cấu trúc và được phân loại bằng đuôi của chính nó. Ví dụ:

File word .docx

File excel .xlsx

File text .txt

File thực thi .exe

- ★ Về mặt kỹ thuật chỉ có 2 loại chính: tập tin văn bản thô và tập tin nhị phân

193  
chu\_trinh\_javac

## Phần 7 : Tập tin



### Tổng quan

- ★ Tập tin văn bản thô (phổ biến nhất là \*.txt) có thể lưu trữ các ký tự. Nó có cấu trúc đơn giản và thông dụng nhất.

- ★ Hầu hết mã nguồn hiện nay đều được viết bằng tập tin văn bản thô

194  
chu\_trinh\_javac

## Phần 7 : Tập tin



### Tổng quan

- ★ Tập tin nhị phân là các tập tin không có cấu trúc giống tập tin văn bản thô mà nó phụ thuộc vào phần mềm sử dụng

- ★ Ví dụ như tập tin âm thanh, hình ảnh, video...

195  
chu\_trinh\_javac

## Phần 7 : Tập tin



### Tổng quan

- ★ Có 3 bước để thao tác với tập tin:

B1: Mở tập tin

Phải có đường dẫn và tên tập tin chính xác

B2: Xử lý tập tin

Đọc hoặc ghi dữ liệu từ tập tin

B3: Đóng tập tin

196  
chu\_trinh\_javac

## Phần 7 : Tập tin



### Khai báo

- ★ Kiểu dữ liệu FILE \*

### Ví dụ

```
FILE *fpt1;  
FILE *fpt2;
```

197  
chu\_trinh\_javac

## Phần 7 : Tập tin



### Một số lệnh cơ bản tập tin văn bản thô

- ★ Đóng mở tập tin

fopen();

fclose();

- ★ Đọc ghi dữ liệu

fscanf(); fgets(); fgetc();

fprintf(); fputs(); fputc();

198  
chu\_trinh\_javac

## Phần 7 : Tập tin



### Một số lệnh cơ bản tập tin văn bản thô

★ Các mode khi đóng mở tập tin văn bản thô

“r”, “rt”: read \_ đọc  
“w”, “wt”: write \_ ghi  
“r+”, “w+”: read write \_ đọc ghi  
“a”, “a+”: mở file đã có sẵn

199  
CHU\_2011\_SEM03

## Phần 7 : Tập tin



### Ví dụ

★ Mở file văn bản thô tên Data.txt và ghi giá trị từ 0 đến 99 vào từng dòng

200  
CHU\_2011\_SEM03

## Phần 7 : Tập tin



### Ví dụ

```
1 <khai báo thư viện>
2 void main()
3 {
4     FILE *fpt1;
5     fpt1=fopen("Data.txt","w"); //Mở file
6     for(int i=0;i<100;i++)
7     {
8         fprintf(fpt1,"%d\n",i); //Ghi file
9     }
10    fclose(fpt1);           //Đóng file
11    printf("Da ghi xong!!!");
12    getch();
13 }
```

201  
CHU\_2011\_SEM03

## Phần 7 : Tập tin



### Ví dụ

★ Mở lại file văn bản thô Data.txt trên và đọc giá trị của nó. Ghi các giá trị vào mảng int a[100]. In mảng ra màn hình.

202  
CHU\_2011\_SEM03

## Phần 7 : Tập tin



### Ví dụ

```
1 <khai báo thư viện>
2 void main()
3 {
4     FILE *fpt1; int a[100];
5     fpt1=fopen("Data.txt","r"); //Mở file
6     for(int i=0;i<100;i++)
7     {
8         fscanf(fpt1,"%d",&a[i]); //Đọc file
9     }
10    fclose(fpt1);           //Đóng file
11    for(int i=0;i<100;i++)
12    {
13        printf("%d\n",a[i]);
14    }
15 }
```

203  
CHU\_2011\_SEM03

## Phần 7 : Tập tin



### Một số lệnh cơ bản tập tin nhị phân

★ Đóng mở tập tin

fopen();  
fclose();

★ Đọc ghi dữ liệu

```
fread(<dịa chỉ biến>, <kích cỡ>, <số khối data>, <con trỏ file>);

fwrite(<dịa chỉ biến>, <kích cỡ>, <số khối data>, <con trỏ file>);
```

204  
CHU\_2011\_SEM03

## Phần 7 : Tập tin



### Một số lệnh cơ bản tập tin nhị phân

★ Các mode khi đóng mở tập tin nhị phân

“rb”: read \_ đọc

“wb”: write \_ ghi

“r+b”, “w+b”: read write \_ đọc ghi

“ab”, “a+b”: mở file đã có sẵn

205  
CHU\_2011\_SEM2

## Phần 7 : Tập tin



### Ví dụ

```
1 <khai báo thư viện>
2 void main()
3 {
4     FILE *fpt1;
5     fpt1=fopen("Data.raw","wb"); //Mở file
6     for(int i=0;i<256;i++)
7         fwrite(&i,1,1,fpt1); //Ghi file
8     fclose(fpt1); //Đóng file
9     printf("Da ghi xong!!!");
10    getch();
11 }
```

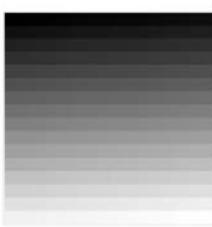
207  
CHU\_2011\_SEM2

## Phần 7 : Tập tin



### Ví dụ

★ Kết quả khi mở file Data.raw bằng Photoshop



209  
CHU\_2011\_SEM2

## Phần 7 : Tập tin



### Ví dụ

★ Mở file nhị phân tên Data.raw và ghi giá trị từ 0 đến 255 liên tiếp nhau

206  
CHU\_2011\_SEM2

## Phần 7 : Tập tin



### Ví dụ

★ Kết quả khi mở file Data.raw bằng HEX Editor

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000010	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
00000020	20	21	22	23	24	25	26	27	28	29	2a	2b	2c	2d	2e	2f
00000030	30	31	32	33	34	35	36	37	38	39	3a	3b	3c	3d	3e	3f
00000040	40	41	42	43	44	45	46	47	48	49	4a	4b	4c	4d	4e	4f
00000050	50	51	52	53	54	55	56	57	58	59	5a	5b	5c	5d	5e	5f
00000060	60	61	62	63	64	65	66	67	68	69	6a	6b	6c	6d	6e	6f
00000070	70	71	72	73	74	75	76	77	78	79	7a	7b	7c	7d	7e	7f
00000080	80	81	82	83	84	85	86	87	88	89	8a	8b	8c	8d	8e	8f
00000090	90	91	92	93	94	95	96	97	98	99	9a	9b	9c	9d	9e	9f
000000a0	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	aa	ab	ac	ad	ae	af
000000b0	b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	ba	bb	bc	bd	be	bf
000000c0	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	ca	cb	cc	cd	ce	cf
000000d0	d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	da	db	dc	dd	de	df
000000e0	e0	e1	e2	e3	e4	e5	e6	e7	e8	e9	ea	eb	ec	ed	ee	ef
000000f0	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	fa	fb	fc	fd	fe	ff

208  
CHU\_2011\_SEM2

## Phần 7 : Tập tin



### Ví dụ

★ Mở lại file nhị phân Data.raw và đọc giá trị của nó. Ghi các giá trị vào mảng unsigned char a[256]. In mảng ra màn hình.

210  
CHU\_2011\_SEM2

## Phần 7 : Tập tin



### Ví dụ

```
1 <khai báo thư viện>
2 void main()
3 {
4     FILE *fpt1; unsigned char a[256];
5     fpt1=fopen("Data.raw","rb"); //Mở file
6     for (int i=0;i<256;i++)
7         fread(&a[i],1,1,fpt1); //Đọc file
8     fclose(fpt1);           //Đóng file
9     for (int i=0;i<256;i++) printf("%d\n",a[i]);
10    getch();
11 }
```

211  
CHU\_7TIN\_BAC1

## Phần 7 : Tập tin



### Một số lệnh cơ bản

- ★ Khi đọc tập tin mà không biết trước số phần tử, có thể dùng hàm feof(<con trỏ file>) để nhận biết kết thúc file

212  
CHU\_7TIN\_BAC1

## Phần 7 : Tập tin



### Bài tập

- ★ Viết chương trình tạo 100 số nguyên dương ngẫu nhiên và ghi vào file SoNguyen.txt
- ★ Viết chương trình đọc file SoNguyen.txt ở trên, sắp xếp mảng số nguyên theo thứ tự tăng dần và ghi vào file SoNguyen\_kq.txt
- ★ Viết chương trình đọc file SoNguyen.txt ở trên, ghi số chẵn vào file SoChan.txt, số lẻ vào file SoLe.txt

213  
CHU\_7TIN\_BAC1

## Phần 7 : Tập tin



### Ghi đọc struct

- ★ Ví dụ struct như sau:

```
typedef struct SinhVien
{
    char MSSV[8];
    char HoTen[30];
    char Khoa[40];
    int Lop;           //1 hoặc 2
};
```

Lưu trữ 2 sinh viên có kiểu struct như trên vào file Data\_sv.dat

214  
CHU\_7TIN\_BAC1

## Phần 7 : Tập tin



### Ví dụ

```
1 <khai báo thư viện>
2 void main()
3 {
4     typedef struct SinhVien
5     {
6         char MSSV[8];
7         char HoTen[30];
8         char Khoa[40];
9         int Lop;           //1 hoặc 2
10    };
11    SinhVien SV1, SV2;
12    strcpy(SV1.MSSV, "1820001");
13    strcpy(SV1.HoTen, "Tran Van A");
14    strcpy(SV1.Khoa, "Dien Tu- Vien Thong");
15    SV1.Lop=1;
16    strcpy(SV2.MSSV, "1820002");
17    strcpy(SV2.HoTen, "Nguyen Thi B");
18    strcpy(SV2.Khoa, "Dien Tu- Vien Thong");
19    SV2.Lop=2;
```

215  
CHU\_7TIN\_BAC1

## Phần 7 : Tập tin



### Ví dụ

```
20
21     FILE *fpt1;
22     fpt1=fopen("Data_sv.dat","wb");
23     fwrite(&SV1,sizeof(SinhVien),1,fpt1);
24     fwrite(&SV2,sizeof(SinhVien),1,fpt1);
25     fclose(fpt1);
26     printf("Da ghi xong!!!");
27 }
```

216  
CHU\_7TIN\_BAC1

## Phần 7 : Tập tin



### Ghi đọc struct

- ★ Lấy lại thông tin 2 sinh viên từ file Data\_sv.dat

217  
CTT\_PPT11\_SEM2

## Phần 7 : Tập tin



### Ví dụ

```
17 printf("Sinh viên 1\nMSSV:%s\nHo Ten sinh viên:%s",
18 SV1.MSSV, SV1.HoTen);
19 printf("\nKhoa:%s\nLop:%d", SV1.Khoa,SV1.Lop);
20 printf("\n\nSinh viên 2\nMSSV:%s\nHo Ten sinh viên:%s",
21 SV2.MSSV, SV2.HoTen);
22 printf("\nKhoa:%s\nLop:%d", SV2.Khoa,SV2.Lop);
23 getch();
24 }
```

219  
CTT\_PPT11\_SEM2

## Phần 7 : Tập tin



### Bài tập

- ★ Viết chương trình quản lý lớp học với 3 lớp, mỗi lớp 10 sinh viên. Mỗi sinh viên gồm có MSSV, Họ Tên, điểm trung bình. Dữ liệu lưu trữ vào file DanhSach.dat
- ★ Xuất danh sách các lớp ra file DanhSach.txt
- ★ Nhập vào mã số, đọc file DanhSach.dat để tìm sinh viên tương ứng
- ★ Đọc file DanhSach.dat để tìm sinh viên có điểm trung bình cao nhất, thấp nhất

221  
CTT\_PPT11\_SEM2

## Phần 7 : Tập tin



### Ví dụ

```
1 <khai báo thư viện>
2 void main()
3 {
4     typedef struct SinhVien
5     {
6         char MSSV[8];
7         char HoTen[30];
8         char Khoa[40];
9         int Lop;           //1 hoặc 2
10    };
11    SinhVien SV1, SV2;
12    FILE *fpt1;
13    fpt1=fopen("Data_sv.dat","rb");
14    fread(&SV1,sizeof(SinhVien),1,fpt1);
15    fread(&SV2,sizeof(SinhVien),1,fpt1);
16    fclose(fpt1);
```

218  
CTT\_PPT11\_SEM2

## Phần 7 : Tập tin



### Ghi đọc struct

- ★ Kết quả

```
Sinh viên 1
MSSV:12345678
Ho Ten sinh viên:Tran Van A
KhaoDien Tu: Van Thong
Lop12

Sinh viên 2
MSSV:18200002
Ho Ten sinh viên:Nguyen Thi B
KhaoDien Tu: Van Thong
Lop12
```

220  
CTT\_PPT11\_SEM2

## Phần 7 : Tập tin



### Bài tập

- ★ Đọc một file hình ảnh grayscale dạng raw độ phân giải bất kỳ
- ★ Tăng độ sáng của hình
- ★ Giảm độ sáng của hình

Lưu ý: Phải có phần mềm Photoshop để tạo/xem hình  
222  
CTT\_PPT11\_SEM2

## Phần 8 : Đệ quy



### Tổng quan

- ★ **Đệ quy (recursion)** là cách gọi 1 hàm trong chính hàm đó
- ★ Khi lập trình sử dụng đệ quy sẽ giúp cho cách viết code đơn giản hơn. Tuy nhiên chương trình sẽ chậm hơn do phải gọi hàm liên tiếp nhiều lần.
- ★ Các bài toán sử dụng phương pháp đệ quy phải có khả năng gọi lại chính nó với tham số nhỏ dần. Có thể viết:  
 $f(n) = f(n-1)....$

223  
CHUYENNGHANH

## Phần 8 : Đệ quy



### Phân loại

- ★ **Đệ quy tuyến tính:** duy nhất 1 lời gọi hàm tường minh
- ★ **Đệ quy nhị phân:** 2 lời gọi hàm tường minh
- ★ **Đệ quy hổ tương:** thân hàm này gọi tới hàm kia và ngược lại
- ★ **Đệ quy phi tuyến:** thân hàm này gọi hàm chính nó bên trong vòng lặp

224  
CHUYENNGHANH

## Phần 8 : Đệ quy



### Ví dụ

- ★ Bài toán  $n!$  ( $n$  giải thừa) có thể được phân tích như sau:  
 $n! = n * (n-1)! = n * (n-1) * (n-2)! = ....$
- ★ **Cấu trúc chung của hàm đệ quy**  
<kiểu dữ liệu> <tên hàm> (<tham số>)  
{  
    if (<điều kiện cuối>) { <trả về giá trị cuối> }  
    <trả về giá trị hiện tại kết hợp với hàm có tham số nhỏ hơn>  
}

225  
CHUYENNGHANH

## Phần 8 : Đệ quy



### Ví dụ

```
1 // Hàm tính giải thừa
2 int GiaiThua(int n)
3 {
4     if (n<=1) return 1;
5     return (n * GiaiThua(n-1));
6 }
```

226  
CHUYENNGHANH

## Phần 8 : Đệ quy



### Bài tập

- ★ Viết hàm trả về giá trị thứ  $n$  của dãy Fibonacci
  - $n=0 \quad F(n)=0$
  - $n=1 \quad F(n)=1$
  - $n=2 \quad F(n)=1$
  - $n=3 \quad F(n)=2$
  - $n=4 \quad F(n)=3$
  - $n=5 \quad F(n)=5$

227  
CHUYENNGHANH

## Phần 8 : Đệ quy



### Bài tập

```
1 // Hàm tính giá trị thứ  $n$  dãy Fibonacci
2 int Fibonacci(int n)
3 {
4     if (n==0 || n==1) return n;
5     return (.....);
6 }
```

228  
CHUYENNGHANH

## Phần 8 : Đệ quy



### Bài tập

★ Viết hàm tính ước số chung lớn nhất  
// Hàm tính ước số chung lớn nhất với m>n  
int USCLN(int m,int n)  
{  
 if (n==0) return m;  
 return USCLN(m.....);  
}

229  
CTT\_PTHI\_BHMS

## Phần 8 : Đệ quy



### Bài tập

★ Viết chương trình nhập vào một số nguyên dương n bất kỳ. Tính biểu thức sau sử dụng phương pháp đệ quy

$$S_n = \frac{1}{1} + \frac{1}{1+2} + \dots + \frac{1}{1+2+3+\dots+n}$$

231  
CTT\_PTHI\_BHMS

## Phần 8 : Đệ quy



### Bài tập

```
1 void chuyen(int n, char A, char C)
2 {
3     printf("Chuyen dia thu %d tu cot %c sang cot
4 %c\n", n, A, C);
5 }
6 void thapHanoi(int n, char A, char B, char C)
7 {
8     if (n==1) chuyen (1,A,C);
9     else
10    {
11        .....;
12        .....;
13        .....;
14    }
15 }
```

233  
CTT\_PTHI\_BHMS

## Phần 8 : Đệ quy



### Bài tập

★ Viết chương trình nhập vào một số nguyên dương n bất kỳ. Tính biểu thức sau sử dụng phương pháp đệ quy

$$S_n = \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$$

230  
CTT\_PTHI\_BHMS

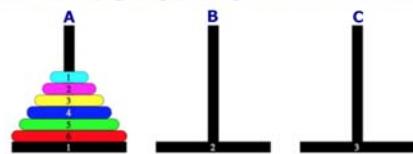
## Phần 8 : Đệ quy



### Bài tập

★ Bài toán tháp Hà Nội

Di chuyển chồng đĩa từ cột A sang cột C mà vẫn giữ nguyên hình dạng tháp (không để đĩa nhỏ dưới đĩa to).



232  
CTT\_PTHI\_BHMS

## Phần 8 : Đệ quy



### Bài tập

```
1 void main()
2 {
3
4
5
6
7
8
9
10
11
12
13
14
15 }
```

234  
CTT\_PTHI\_BHMS

## Phụ lục 1 : Các hệ cơ số đếm



### Các hệ cơ số đếm

- ★ Thập phân (decimal) : con người sử dụng
- ★ Nhị phân (binary) : dùng trong các hệ thống số
- ★ Thập lục phân (hexadecimal) và bát phân (octal) : hỗ trợ biểu diễn hệ nhị phân

235  
CPU\_CHIP\_IMAGES

## Phụ lục 1 : Các hệ cơ số đếm



### Các hệ cơ số đếm

- ★ Hệ thống đếm thập phân (Decimal)

Cơ số: 10

Các chữ số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Biểu diễn tổng quát một số

$$A_{10} = a_n a_{n-1} \dots a_1 a_0 = \\ = a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_1 10^1 + a_0 10^0$$

236  
CPU\_CHIP\_IMAGES

## Phụ lục 1 : Các hệ cơ số đếm



### Các hệ cơ số đếm

- ★ Hệ thống đếm nhị phân (Binary)

Cơ số: 2

Các chữ số: 0, 1

Biểu diễn tổng quát một số

$$A_2 = a_n a_{n-1} \dots a_1 a_0 = \\ = a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_1 2^1 + a_0 2^0$$

237  
CPU\_CHIP\_IMAGES

## Phụ lục 1 : Các hệ cơ số đếm



### Các hệ cơ số đếm

- ★ Hệ thống đếm bát phân và thập lục phân

Cơ số: 8 và 16

Các chữ số hệ bát phân : 0, 1, 2, 3, 4, 5, 6, 7

Các chữ số hệ thập lục phân : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

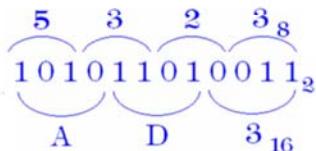
238  
CPU\_CHIP\_IMAGES

## Phụ lục 1 : Các hệ cơ số đếm



### Các hệ cơ số đếm

- ★ Cách chuyển đổi từ nhị phân sang bát phân và thập lục phân



239  
CPU\_CHIP\_IMAGES

## Phụ lục 1 : Các hệ cơ số đếm



### Bài tập

$$\star 13 \rightarrow \dots \dots \dots_2 \rightarrow \dots \dots \dots_8 \rightarrow \dots \dots \dots_{16}$$

$$59 \rightarrow \dots \dots \dots_2 \rightarrow \dots \dots \dots_8 \rightarrow \dots \dots \dots_{16}$$

$$90 \rightarrow \dots \dots \dots_2 \rightarrow \dots \dots \dots_8 \rightarrow \dots \dots \dots_{16}$$

$$\star 1111_2 \rightarrow \dots \dots \dots_{10} \rightarrow \dots \dots \dots_8 \rightarrow \dots \dots \dots_{16}$$

$$110111_2 \rightarrow \dots \dots \dots_{10} \rightarrow \dots \dots \dots_8 \rightarrow \dots \dots \dots_{16}$$

$$1010101_2 \rightarrow \dots \dots \dots_{10} \rightarrow \dots \dots \dots_8 \rightarrow \dots \dots \dots_{16}$$

240  
CPU\_CHIP\_IMAGES

## Phụ lục 1 : Các hệ cơ số đếm



### Hệ nhị phân (Binary)

- ★ Cách biểu diễn một số âm (số có dấu)

Sử dụng số bù 2 (bù 1 + 1)

Đối với số có dấu : bit MSB của số dương = 0

bit MSB của số âm = 1

0101011100  
MSB                    LSB

241

## Phụ lục 1 : Các hệ cơ số đếm



### Bài tập

- ★ Cách biểu diễn một số âm (số có dấu)

$$\begin{array}{rcl}
 ?_{10} = 11011_2 & \xleftarrow{\text{bù 2}} & ?_2 = ?_{10} \\
 ?_{10} = 11111_2 & \xleftarrow{\text{bù 2}} & ?_2 = ?_{10} \\
 ?_{10} = ?_2 & \xleftarrow{\text{bù 2}} & 01101_2 = ?_{10} \\
 -12_{10} = ?_2 & \xleftarrow{\text{bù 2}} & ?_2 = +12_{10} \\
 -35_{10} = ?_2 & \xleftarrow{\text{bù 2}} & ?_2 = +35_{10} \\
 ?_{10} = 10000_2 & \xleftarrow{\text{bù 2}} & ?_2 = ?_{10}
 \end{array}$$

243

## Phụ lục 2 : Tìm kiếm - Sắp xếp



### Tìm kiếm (search)

- ★ Thao tác tìm kiếm khá phổ biến trong thực tế nhằm khai thác thông tin

- ★ Các hệ thống thông tin thường lưu trữ một khối lượng dữ liệu rất lớn, do đó các giải thuật tìm kiếm nhanh sẽ có ý nghĩa rất lớn

- ★ Ví dụ: tra cứu từ điển, tìm thông tin lưu trữ

245

## Phụ lục 1 : Các hệ cơ số đếm



### Hệ nhị phân (Binary)

- ★ Cách biểu diễn một số âm (số có dấu)

$$\begin{array}{rccccc}
 1001_2 & \xrightarrow{\text{bù 1}} & 0110_2 & \xrightarrow{\text{bù 2}} & 0111_2 \\
 1100_2 & \xrightarrow{\text{bù 1}} & 0011_2 & \xrightarrow{\text{bù 2}} & 0100_2
 \end{array}$$

242

## Phụ lục 1 : Các hệ cơ số đếm



### Bài tập : Biểu diễn số nguyên có dấu

- ★  $-13 \rightarrow \dots_2 \rightarrow \dots_8 \rightarrow \dots_{16}$  (có dấu)
- $-59 \rightarrow \dots_2 \rightarrow \dots_8 \rightarrow \dots_{16}$  (có dấu)
- $+19 \rightarrow \dots_2 \rightarrow \dots_8 \rightarrow \dots_{16}$  (có dấu)
- ★  $1101_2 \rightarrow \dots_{10} \rightarrow \dots_8 \rightarrow \dots_{16}$  (có dấu)
- $0111_2 \rightarrow \dots_{10} \rightarrow \dots_8 \rightarrow \dots_{16}$  (có dấu)
- $10101_2 \rightarrow \dots_{10} \rightarrow \dots_8 \rightarrow \dots_{16}$  (có dấu)

244

## Phụ lục 2 : Tìm kiếm - Sắp xếp



### Tìm kiếm (search)

- ★ Có 2 giải thuật tìm kiếm thông dụng là tìm kiếm tuyến tính và tìm kiếm nhị phân

- ★ Tìm kiếm tuyến tính là kỹ thuật đơn giản nhất, nó so sánh giá trị cần tìm trong một mảng một cách lần lượt từng phần tử cho đến khi tìm thấy

```

int TimKiem(int a[], int N, int x)
{
    for (int i=0, i<N, i++) if (a[i]==x) return i;
    if (i==N-1) return -1; //không tìm thấy
}

```

246

## Phụ lục 2 : Tìm kiếm - Sắp xếp



### Tìm kiếm (search)

- Đối với mảng đã có thứ tự (nhỏ dần/lớn dần) thì dùng cách tìm kiếm nhị phân
- Tìm kiếm nhị phân: mỗi lần sẽ tìm kiếm tại vị trí giữa của mảng đang tìm, từ đó sẽ giới hạn phân nửa mảng đang tìm

247  
CHU\_2011\_BNEWS

## Phụ lục 2 : Tìm kiếm - Sắp xếp



### Tìm kiếm (search)

```
// Hàm tìm kiếm nhị phân với mảng tăng dần
int TimKiem(int a[], int N, int x)
{
    int trai=0; int phai=N-1; int gieu;
    do {
        gieu = (trai + phai)/2;
        if (x==a[gieu]) return gieu; //tim thấy
        if (x<a[gieu]) phai=gieu-1;
        else trai = gieu+1;
    } while (trai<=phai);
    return -1; //không tìm thấy
}
```

248  
CHU\_2011\_BNEWS

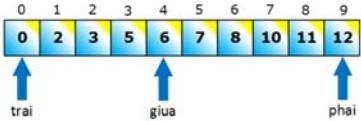
## Phụ lục 2 : Tìm kiếm - Sắp xếp



### Tìm kiếm (search)

- Ví dụ tìm kiếm nhị phân: tìm giá trị 7 trong mảng 10 phần tử

Bước 1



249  
CHU\_2011\_BNEWS

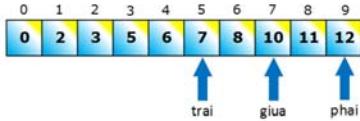
## Phụ lục 2 : Tìm kiếm - Sắp xếp



### Tìm kiếm (search)

- Ví dụ tìm kiếm nhị phân: tìm giá trị 7 trong mảng 10 phần tử

Bước 2



250  
CHU\_2011\_BNEWS

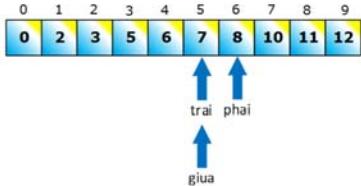
## Phụ lục 2 : Tìm kiếm - Sắp xếp



### Tìm kiếm (search)

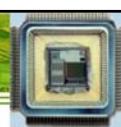
- Ví dụ tìm kiếm nhị phân: tìm giá trị 7 trong mảng 10 phần tử

Bước 3



251  
CHU\_2011\_BNEWS

## Phụ lục 2 : Tìm kiếm - Sắp xếp



### Sắp xếp (sort)

- Như vậy để tăng hiệu quả tìm kiếm trong mảng thì mảng đó phải được sắp xếp theo thứ tự nhất định

- Có rất nhiều thuật toán sắp xếp khác nhau

Selection sort

Interchange sort

Bubble sort

Quick sort

Merge sort.....

252  
CHU\_2011\_BNEWS





### Hướng đối tượng

- ★ Mỗi lớp bao gồm các thành phần chính như sau: tên, thuộc tính, phương thức
- ★ Quan hệ giữa các đối tượng là ngang hàng hoặc kế thừa: lớp B kế thừa từ lớp A, thì lớp A là lớp cơ sở (cha), lớp B là lớp dẫn xuất (con)

259  
CTT\_PPT1\_SEM05



### Hướng đối tượng

- ★ 2 khái niệm quan trọng tạo nên thế mạnh của OOP là tính kế thừa (inheritance) và tính đa hình (polymorphism)
- ★ Tính kế thừa cho phép xây dựng một lớp mới dựa trên lớp đã có. Lớp mới gọi là lớp con kế thừa toàn bộ các thành phần của lớp cha, có thể chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại

260  
CTT\_PPT1\_SEM05



### Hướng đối tượng

- ★ Tính đa hình cho phép các đối tượng thuộc các lớp khác nhau có cùng lớp cha có thể thực hiện tác vụ theo những cách khác nhau
- ★ Nói một cách khác tính đa hình cho phép định nghĩa lại tác vụ ở lớp con mà lớp cha đã có sẵn

261  
CTT\_PPT1\_SEM05



### POP vs OOP

- ★ Khi viết các chương trình nhỏ thì dùng phương pháp hướng cấu trúc. Ngược lại phương pháp hướng đối tượng phù hợp cho các chương trình lớn, độ phức tạp cao.

262  
CTT\_PPT1\_SEM05