

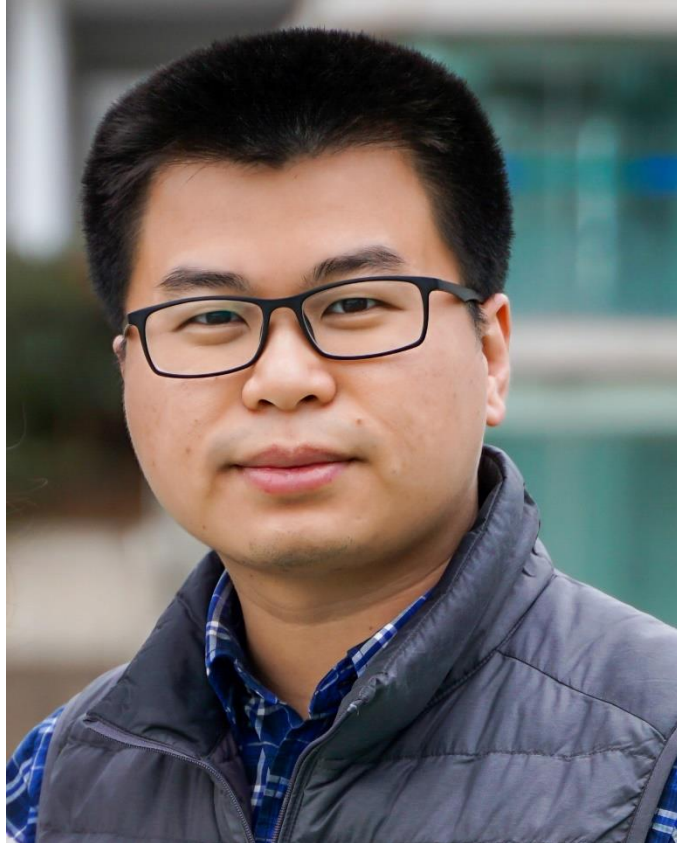


# LẬP TRÌNH NÂNG CAO

GIẢNG VIÊN

TS. Hà Ngọc Long

# THÔNG TIN GIẢNG VIÊN



**TS. Hà Ngọc Long**

Email: [hnlng@hueuni.edu.vn](mailto:hnlng@hueuni.edu.vn)

- Thông tin Giảng viên
  - Tiến sĩ ngành Hệ thống Thông tin và Truyền thông, Hàn Quốc.
  - Thạc sĩ ngành Kỹ thuật Quản trị Công nghiệp, Hàn Quốc.
  - Cử nhân ngành Hệ thống Thông tin Quản lý, Đại học Kinh tế Huế
- Hướng nghiên cứu:
  - Quản trị quy trình nghiệp vụ (Business Process Management).
  - Chuyển đổi số.
  - FinTech, RegTech, AI
  - Tiền điện tử, hợp đồng thông minh, và Blockchain.
  - Khai phá quy trình nghiệp vụ (Process Mining)

# MỤC TIÊU CỦA HỌC PHẦN

**“Giới thiệu về ngôn ngữ lập trình C. Tập trung giới thiệu cho sinh viên kỹ năng, kỹ thuật lập trình theo phương pháp hiện đại”**

# TIÊU CHUẨN ĐÁNH GIÁ SINH VIÊN

Hình thức Đánh giá	% Điểm
Điểm danh	15%
Kiểm tra trên máy tính 1	15%
Kiểm tra trên máy tính 2	20%
<b><i>Thi kết thúc học phần</i></b> <i>(Làm bài trên máy tính)</i>	<b>50%</b>
<b>Tổng cộng:</b>	<b>100%</b>

# TÀI LIỆU HỌC TẬP

- Trương Công Tuấn. Ngôn ngữ lập trình C – Các vấn đề cốt yếu, NXB Đại học Huế, 2008.
- Nguyễn Thanh Thủy. Ngôn ngữ C. NXB Khoa học và Kỹ thuật, 1999.
- Phạm Văn Ất. Kỹ thuật lập trình C. NXB Khoa học và Kỹ thuật, 1995.

Tài liệu hướng dẫn và phần mềm thực hành:

- Dev-C++ (IDE)
- Visual Studio Code (IDE)
- Videos bài giảng và hướng dẫn thực hành

# NỘI DUNG MÔN HỌC

1

- Kiểu Chuỗi

2

- Kiểu Cấu Trúc

3

- Kiểu Tập Tin

# Hình thức tổ chức lớp học

- Sinh viên nghe giảng và thực hành đối với *Chương 1* và *Chương 2*
- **Hoạt động nhóm:**
  - Sinh viên chia thành những nhóm nhỏ, tìm hiểu và trình bày về những chủ đề lập trình được giao (Cuối *Chương 2* và *Chương 3*).
  - Hoạt động nhóm sẽ được tính vào một cột điểm kiểm tra giữa kỳ.



# CHƯƠNG 1

## KIỂU CHUỖI

GIẢNG VIÊN

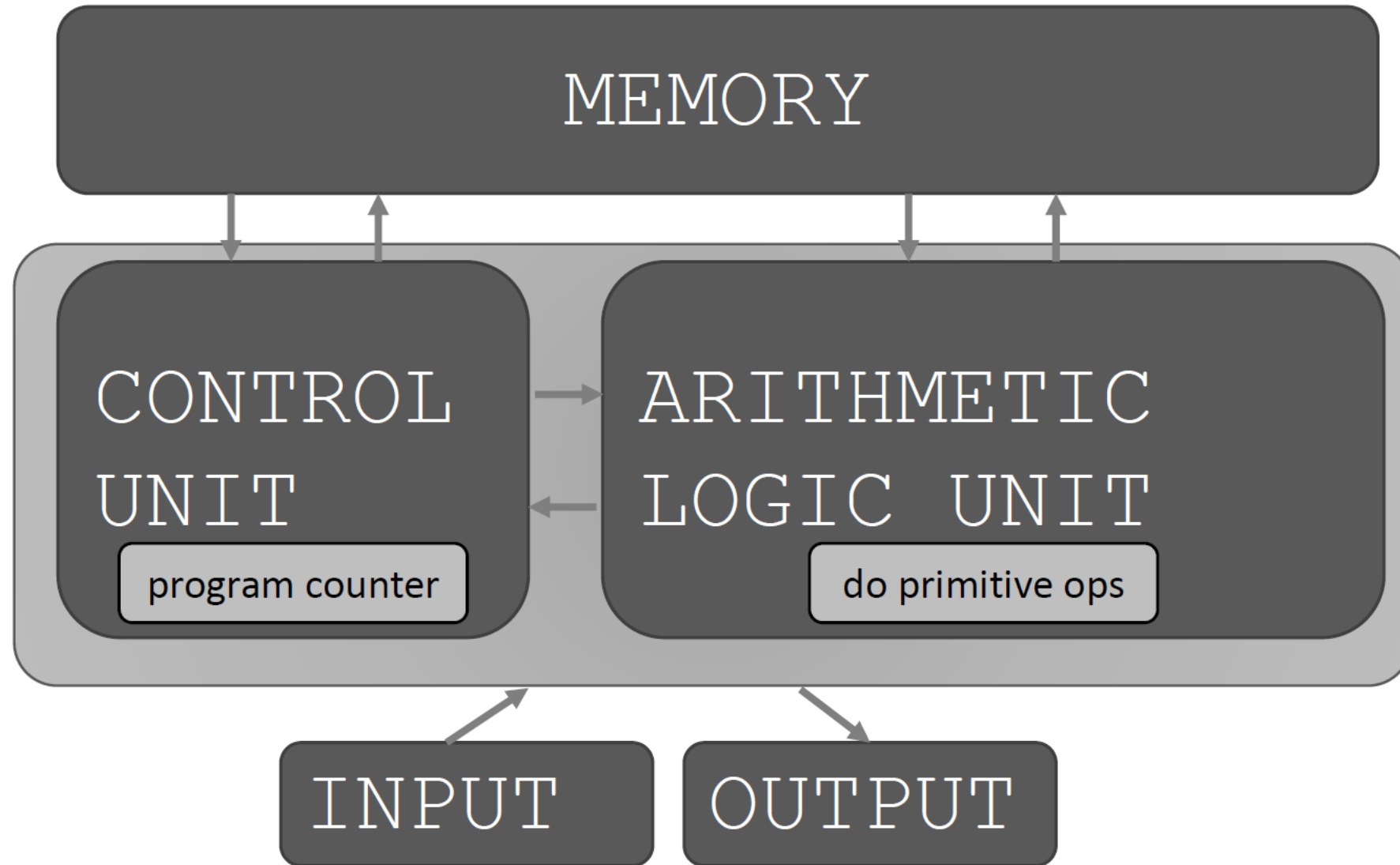
TS. Hà Ngọc Long





# Quy Trình Giải Bài Toán & Lập Trình

# Kiến trúc cơ bản của máy tính



# Dữ liệu được lưu trữ như thế nào?

- Dữ liệu (Data) có nhiều dạng khác nhau, như số, ký tự, chuỗi ký tự. Chúng thường được mã hóa dưới dạng một chuỗi các bit (0 và 1).
- Máy tính sử dụng 0 và 1 bởi vì thiết bị điện tử có 2 trạng thái ổn định (có điện và không có điện).
- Lập trình viên không cần quan tâm nhiều đến việc mã hóa và giải mã dữ liệu (hệ thống sẽ hỗ trợ việc này).
- Ví dụ, chữ 'J' được biểu diễn dưới dạng 01001010 trong 1 byte. Nếu dữ liệu cần nhiều hơn 1 byte, máy tính sẽ sử dụng những byte liên kề.

Memory address		Memory content	
.	↓	.	
.		.	
.		.	
2000		01001010	Encoding for character 'J'
2001		01100001	Encoding for character 'a'
2002		01110110	Encoding for character 'v'
2003		01100001	Encoding for character 'a'
2004		00000011	Encoding for number 3

# Giải Bài Toán Bằng Máy Tính

- Giải quyết một bài toán:
  - Làm gì ?
  - Làm như thế nào ?
- Giải quyết Bài toán Tin học phải:
  - Tổ chức biểu diễn các đối tượng thực tế
  - Xây dựng trình tự các thao tác xử lý trên các đối tượng dữ liệu đó

# Giải Bài Toán Bằng Máy Tính

- Hai yếu tố tạo nên một chương trình máy tính
  - Cấu trúc dữ liệu
  - Giải thuật

**Cấu trúc dữ liệu** + **Giải thuật** = **Chương trình**

# Quy Trình Giải Bài Toán (1/9)

## ❑ BƯỚC 1: Xác định bài toán

- Xác định vấn đề cần giải quyết là gì, những giả thiết nào đã có
- Tập trung vào dữ liệu đầu vào (input) và dữ liệu đầu ra (output)



# Quy Trình Giải Bài Toán (2/9)

## ❑ BƯỚC 2: Tìm cấu trúc dữ liệu biểu diễn thuật toán

- Chọn kiểu dữ liệu, hoặc cấu trúc dữ liệu phụ thuộc vào yêu cầu bài toán
- Một số giải thuật chỉ thích ứng với một cách tổ chức dữ liệu nhất định
- Tiêu chuẩn lựa chọn cấu trúc dữ liệu (CTDL):
  - CTDL phải biểu diễn được đầy đủ thông tin nhập (input) và xuất (output) của bài toán.
  - CTDL phải phù hợp với các thao tác của giải thuật được lựa chọn.
  - CTDL phải cài đặt được trên máy tính với Ngôn Ngữ LT đang sử dụng.

# Quy Trình Giải Bài Toán (3/9)

## ❑ BƯỚC 3: Tìm Giải Thuật

- Giải thuật là một hệ thống chặt chẽ và rõ ràng các quy tắc nhằm xác định một dãy thao tác trên cấu trúc dữ liệu sao cho:
  - Với một bộ dữ liệu vào, sau số hữu hạn bước thực hiện các thao tác đã chỉ ra, ta đạt được mục tiêu đã định.



# Quy Trình Giải Bài Toán (4/9)

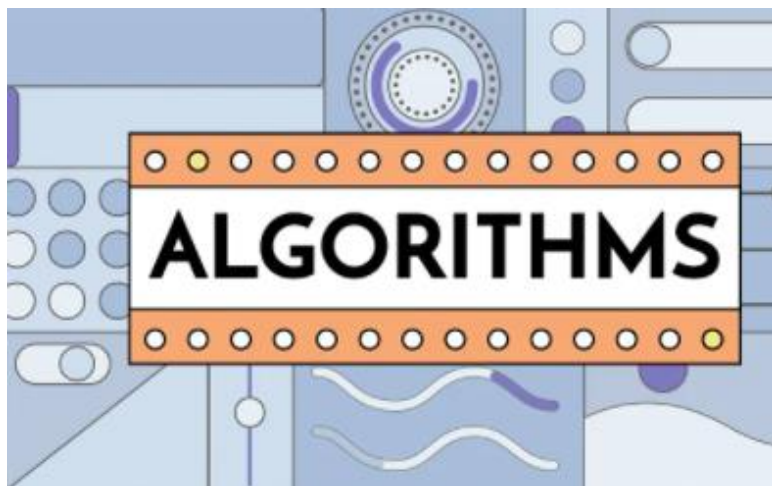
- **Định nghĩa giải thuật:** là dãy các câu lệnh chặt chẽ và rõ ràng xác định một trình tự các thao tác trên một số đối tượng nào đó, sao cho sau một số hữu hạn bước thực hiện ta đạt được kết quả mong muốn
- Mỗi thuật toán có **một dữ liệu vào** (Input) và **một dữ liệu ra** (Output);
- Lý thuyết giải thuật quan tâm đến những vấn đề sau :
  - 1. Giải được bằng giải thuật :
  - 2. Tối ưu hóa giải thuật :
  - 3. Triển khai giải thuật:

# Quy Trình Giải Bài Toán (5/9)



- [https://www.youtube.com/watch?v=oRkNaF0QvnI&ab\\_channel=BBCIdeas](https://www.youtube.com/watch?v=oRkNaF0QvnI&ab_channel=BBCIdeas)

**Nguồn gốc giải thuật**



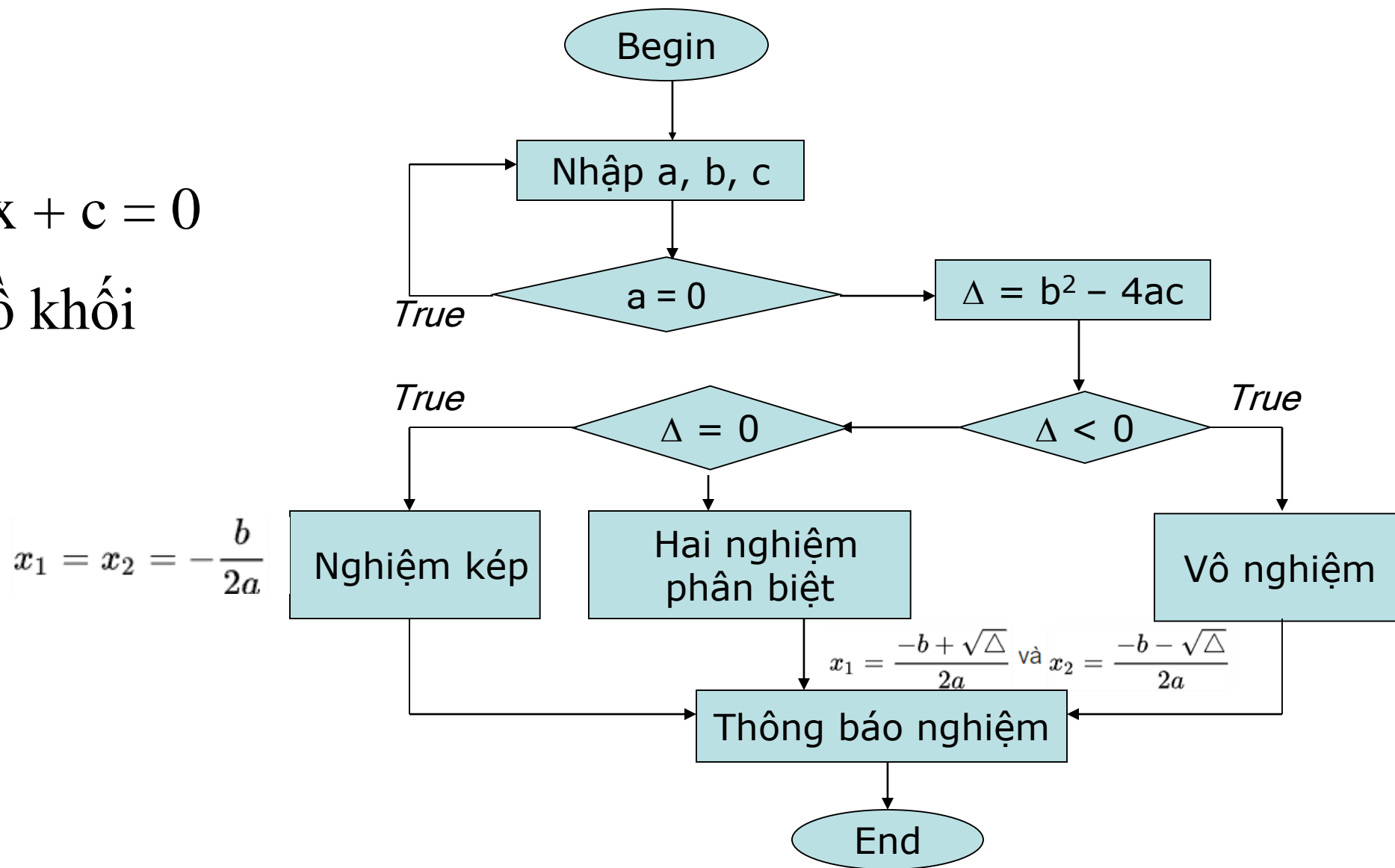
- [https://www.youtube.com/watch?v=kM9ASKAni\\_s&ab\\_channel=GCFLearnFree.org](https://www.youtube.com/watch?v=kM9ASKAni_s&ab_channel=GCFLearnFree.org)

# Quy Trình Giải Bài Toán (6/9) – Ví dụ

- Ví dụ :

Giải PT:  $ax^2 + bx + c = 0$

Mô tả bằng sơ đồ khối



# Quy Trình Giải Bài Toán (7/9) - Diễn Đạt GT

- **Ví dụ 1:** Giải thuật xác định  $n$  là số nguyên tố

- Bước 1: Ghi nhận  $n$
- Bước 2: Nếu  $n \leq 1 \rightarrow n$  ko nguyên tố  $\rightarrow$  dừng
- Bước 3: Nếu  $n > 2$ , gán  $i \rightarrow 2$
- Bước 4: Nếu  $i \geq \sqrt{n}$  hay  $n$  chia hết cho  $i \rightarrow$  bước 6
- Bước 5: Gán  $i \rightarrow i+1$ , trở lại bước 4
- Bước 6:
  - Nếu  $i > \sqrt{n} \rightarrow n$  nguyên tố  $\rightarrow$  dừng
  - Ngược lại,  $n$  không là nguyên tố  $\rightarrow$  dừng

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

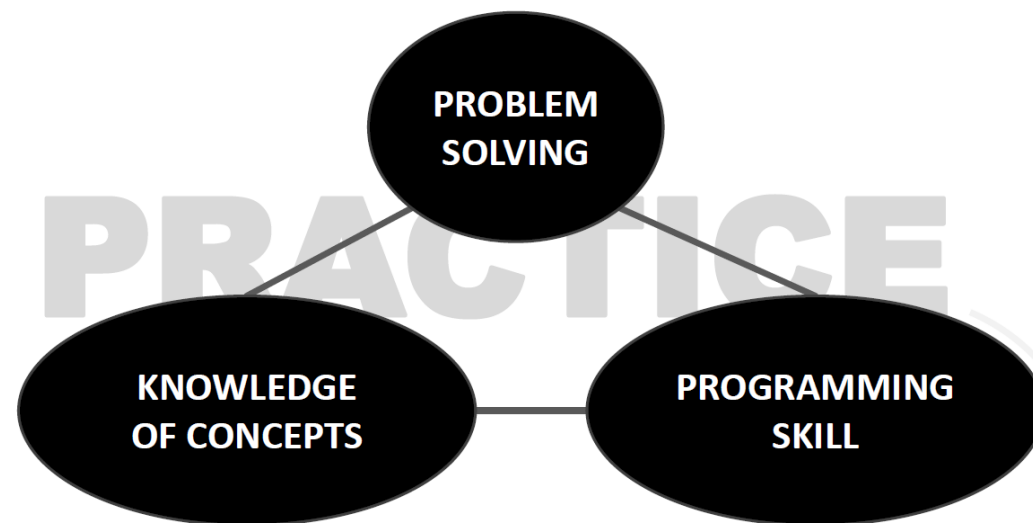
# Quy Trình Giải Bài Toán (8/9) - Chương trình

- Chương trình (program) máy tính, hay còn được gọi là phần mềm, là những chỉ dẫn cho máy tính.
- Sử dụng programs, chúng ta có thể sai khiến máy tính làm việc. Không có program, máy tính có thể xem như bộ máy trống rỗng.
- Máy tính không hiểu được ngôn ngữ con người, vì vậy chúng ta sử dụng ngôn ngữ máy tính để giao tiếp với nó.
- Programs được viết bằng những ngôn ngữ lập trình.

# Quy Trình Giải Bài Toán (9/9) - Lập trình

- **BƯỚC 4: Lập Trình (Programming)**

- Sau khi đã có thuật toán ta phải lập trình để thực hiện thuật toán đó.
- Muốn lập trình đạt hiệu quả cao, cần phải có kỹ thuật lập trình tốt
- New to programming? **PRACTICE. PRACTICE? PRACTICE!**





# Căn Bản Về Chuỗi Ký Tự

# Khái niệm chuỗi ký tự

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

<b>'T'</b>	<b>'i'</b>	<b>'n'</b>	<b>' '</b>	<b>'h'</b>	<b>'o'</b>	<b>'c'</b>	<b>'\0'</b>
------------	------------	------------	------------	------------	------------	------------	-------------



# Lưu ý

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

# Khai báo

```
char tên_xâu [số_kí_tự_tối_đa];
```

- Để lưu trữ một xâu có **n** kí tự chúng ta cần một mảng có kích thước **n+1**
  - Phần tử cuối cùng chứa ký tự NULL
- *Ví dụ*
- • Để lưu trữ xâu “Tin hoc” chúng ta phải khai báo xâu có số phần tử tối đa ít nhất là 8

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

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

# Truy cập phần tử của chuỗi

- Giống như truy nhập tới một phần tử của mảng ký tự.

***tên\_xâu [chỉ\_số\_của\_kí\_tự]***

- Ví dụ:* char Str[10] = “Tin hoc”

<b>T</b>	<b>i</b>	<b>n</b>	<b>-</b>	<b>h</b>	<b>o</b>	<b>c</b>	<b>\0</b>	<b>?</b>	<b>\0</b>
----------	----------	----------	----------	----------	----------	----------	-----------	----------	-----------

Str[0] → 'T'

Str[3] = '-' ;

Str[3] → ' '

Str[7] = ' ' ;

Str[7] → '\\0 '

Str[8] = '\\1 ' ;

Str[8] → ?

Str[9] = '\\0' ;

Str: Tin-hoc 1

# Ví dụ: Nhập chuỗi và đếm số ký tự '\*'

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

Tính chiều dài của chuỗi  
d=0;  
while(Str[d] != '\0') d++;

Nhap xau ky tu: &&\*\*\*\*\*&&&&&\*&  
Ket qua : 8

# TẬP TIÊU ĐỀ ctype.h

```
#include<ctype.h>
```

# Các hàm xử lý ký tự: Chuyển đổi chữ hoa/thường

**toupper(char ch)**

- Chuyển ký tự thường thành ký tự hoa `toupper('a') => 'A'`

**int tolower(char ch)**

- Chuyển ký tự hoa thành ký tự thường `tolower('B') => 'b'`

## Ví dụ

```
do {
```

```
    .....
```

```
    printf("Tiep tuc <C/K>? :"); fflush(stdin);
```

```
} while (toupper(getche()) != 'K');
```

# Các hàm xử lý ký tự: Kiểm tra chữ hoa/thường

**int islower(char ch)**

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

**int isupper(char ch)**

- Kiểm tra chữ hoa:
  - Hàm trả về giá trị khác 0 nếu ch là chữ hoa, ngược lại trả về 0
  - Ví dụ: `printf("%d ", isupper('A')) ;`  $\Rightarrow \neq 0$  (1 !?)

# Các hàm xử lý ký tự: Kiểm tra chữ cái/chữ số

## **int isalpha(char ch)**

- Kiểm tra ký tự trong tham số có phải chữ cái không ('a'...'z','A',..'Z').  
Hàm trả về khác 0 nếu đúng, ngược lại trả về giá trị bằng 0
- Ví dụ: `printf("%d ", isalpha('A'))`;  $\Rightarrow \neq 0$  (1 !?)

## **int isdigit(char ch)**

- Kiểm tra ký tự trong tham số có phải chữ số ('0','1',..'9') không. Hàm trả về khác 0 nếu đúng, ngược lại trả về giá trị bằng 0
- Ví dụ: `printf("%d ", isdigit('A'))`;  $\Rightarrow 0$



# Khái niệm chuỗi ký tự: Kiểm tra ký tự đặc biệt

**int iscntrl(char ch)**

- Kiểm tra ký tự điều khiển (0-31).
- Hàm trả về khác 0 nếu đúng, ngược lại trả về giá trị bằng 0

**int isspace(char ch)**

- Kiểm tra ký tự dấu cách (mã 32), xuống dòng ( `'\n'` 10), đầu dòng ( `'\r'` 13), tab ngang ( `'\t'` 9), tab dọc ( `'\v'` 11) .
- Hàm trả về khác 0 nếu đúng, ngược lại trả về giá trị bằng 0

# Ví dụ: Nhập chuỗi và đếm từ, phân cách bởi khoảng trắng

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

Nhap xau ky tu: NHAP MON LAP TRINH C  
Ket qua : 5

# Các hàm xử lý chuỗi ký tự

## Vào/ra chuỗi (chuỗi) ký tự

- Tập tiêu đề: `stdio.h`
- Nhập chuỗi ký tự
  - `gets(tên_xâu);`
  - `scanf("%s", &tên_xâu);`
- • Hiển thị chuỗi ký tự
  - `puts(tên_xâu);`
  - `printf("%s", tên_xâu);`

**Sự khác nhau giữa `gets` và `scanf`?**

# TẬP TIÊU ĐỀ string.h

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

# Các hàm xử lý chuỗi ký tự (1/6)

- Chú ý:

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

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

- p là con trỏ tới mảng các ký tự/ chuỗi ký tự
  - p+6 cũng là chuỗi ký tự : world
- Chuỗi ký tự, có thể được khai báo `char *`

# Các hàm xử lý chuỗi ký tự (2/6)

**size\_t strlen(char \* xâu)**

- Trả về độ dài xâu

```
printf("%d ", strlen("Hello world")); ⇒ 11
```

**char \* strcpy(char \* đích, char \* nguồn)**

- Sao chép xâu, trả về giá trị xâu nguồn

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

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

```
printf("%s", Str); ⇒ Hello
```

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

# Các hàm xử lý chuỗi ký tự (3/6)

**int strcmp(char \* xâu\_1, char \* xâu\_2)**

- So sánh hai xâu.
- Trả về giá trị 0 nếu hai xâu giống nhau;
- Giá trị < 0: xâu\_1 < xâu\_2
- Giá trị > 0: xâu\_1 > xâu\_2

- **Ví dụ**

```
char Str[20];  
strcpy(Str, "hello");  
printf("%d", strcmp(Str, "hello")); → 0  
printf("%d", strcmp(Str, "hello!")); → -1 (!?)  
printf("%d", strcmp(Str, "Hello")); → 1 (!?)
```

# Các hàm xử lý chuỗi ký tự (4/6)

**char \* strcat(char \* xđích, char \* nguồn)**

– Ghép nối xâu nguồn vào ngay sau xâu đích, trả lại xâu kết quả

- Ví dụ

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

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

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

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



# Các hàm xử lý chuỗi ký tự (5/6)

**char \* strchr (char \* s, int c)**

- Trả về con trỏ trỏ tới vị trí xuất hiện đầu tiên của ký tự **c** trong **s**. Nếu không có trả về con trỏ null

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

```
printf("%s ", strchr(Str, 'o')); ⇒ o world
```

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

- Trả về con trỏ trỏ tới vị trí xuất hiện đầu tiên của chuỗi **s2** trong **s1**. Nếu không tồn tại, trả về con trỏ null

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

# Các hàm xử lý chuỗi ký tự (6/6)

Tập tiêu đề: **stdlib.h**

**int atoi(char \* str)**

- Chuyển một xâu ký tự thành một số nguyên tương ứng
- Ví dụ: `atoi("1234")` → 1234

**int atol(char \* str)**

- Chuyển xâu ký tự thành số long int

**float atof(char \* str) :**

- Chuyển xâu ký tự thành số thực
- Ví dụ: `atof("123.456E-2")` → 1.23456

**Thất bại cả 3 hàm: trả về 0**

# Ví dụ 1: Đảo ngược chuỗi ký tự

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

Nhap xau: CONG NGHE THONG TIN  
NIT GNOHT EHGN GNOC

## Ví dụ 2: Kiểm tra chuỗi đối xứng

```
#include<stdio.h>
#include<string.h>
main(){
    char s[20];
    int i,n;
    printf("Nhap vao xau ki tu: "); gets(s);
    n=strlen(s);
    for(i=0;i<n/2;i++)
        if(s[i]!=s[n-1-i])
            break;
    if(i==n/2)
        printf("Xau doi xung");
    else
        printf("Xau khong doi xung");
}
```

Nhap vao xau ki tu: ABCCBA  
Xau doi xung

Nhap vao xau ki tu: ABCDDCAA  
Xau khong doi xung

# Ví dụ 3: Đếm số lần xuất hiện chữ cái trong chuỗi

```
#include<stdio.h>
#include<ctype.h>
#include<string.h>
main(){
    char s[20];
    int dem[26] = {};
    int i,n;
    puts("Nhap vao xau ki tu:");gets(s);
    n=strlen(s);
    for(i=0;i<n;i++)
        if(isalpha(s[i]))
            dem[ tolower(s[i]) - 'a' ]++;
    for(i=0;i<26;i++)
        if(dem[i]!=0)
            printf("Ki tu %c xuất hiện %d lần\n",'a'+i,dem[i]);
}
```

```
Nhap vao xau ki tu:
NHAP MON LAP TRINH
Ki tu a xuất hiện 2 lần
Ki tu h xuất hiện 2 lần
Ki tu i xuất hiện 1 lần
Ki tu l xuất hiện 1 lần
Ki tu m xuất hiện 1 lần
Ki tu n xuất hiện 3 lần
Ki tu o xuất hiện 1 lần
Ki tu p xuất hiện 2 lần
Ki tu r xuất hiện 1 lần
Ki tu t xuất hiện 1 lần
```

# Mảng chuỗi ký tự

- Xâu (Chuỗi) ký tự **có thể là kiểu phần tử của mảng**

- **Khai báo**

```
char DS[100][30];
```

Mảng có tối đa 100 phần tử, các phần tử là xâu có độ dài tối đa 30

- **Sử dụng**

- Như một mảng bình thường
- Mỗi phần tử mảng được sử dụng như một xâu ký tự

# Ví dụ 1: Nhập vào DSSV tới khi tên rỗng & in DS

```
#include <stdio.h>
void main(){
    int i, n;
    char DS[100][30];
    printf("Nhap DSSV (<100), go Enter de thoat..\n");
    n =0;
    do{
        printf("Ten sinh vien[%d]: ",n+1); gets(DS[n]);
        if(DS[n][0] != '\x00') n++;
        else break;
        if(n==100) break;
    }while(1);
    printf("\n\nDS sinh vien vua nhap \n");
    for(i=0;i<n;i++) printf("%s\n",DS[i]);
}
```

```
Nhap DSSV (<100), go Enter de thoat..
Ten sinh vien[1]: NGUYEN VAN A
Ten sinh vien[2]: TRAN THI B
Ten sinh vien[3]: LE VAN C
Ten sinh vien[4]: NGUYEN THI D
Ten sinh vien[5]:
```

```
DS sinh vien vua nhap
NGUYEN VAN A
TRAN THI B
LE VAN C
NGUYEN THI D
```

# Ví dụ 2: Nhập vào DSSV & in DS sau sắp xếp

```
#include <stdio.h>
#include <string.h>
void main(){
    int i, j, N;
    char DS[100][30], str[30];
    //Nhap DS lop
    printf("So sinh vien : ");
    scanf("%d",&N);
    fflush(stdin);
    for(i=0;i < N;i++){
        printf("Ten sinh vien[%d]: ",i);
        gets(DS[i]);
    }
```

```
So sinh vien : 4
Ten sinh vien[0]: NGUYEN VAN C
Ten sinh vien[1]: LE VAN A
Ten sinh vien[2]: TRAN THI D
Ten sinh vien[3]: LE THI F

DS sinh vien vua nhap
LE THI F
LE VAN A
NGUYEN VAN C
TRAN THI D
```

```
//So sánh theo Họ+đệm+tên
for(i = 0; i < N - 1; i ++){
    for(j = i + 1; j < N; j ++){
        if(strcmp(DS[i],DS[j]) > 0){
            strcpy(str,DS[i]);
            strcpy(DS[i],DS[j]);
            strcpy(DS[j],str);
        }
    }
}

//In danh sach da sắp xếp
printf("\nDS sinh vien vua nhap \n");
for(i=0;i < N;i++){
    printf("%s\n",DS[i]);
}

} //main
```



## Ví dụ 2: Sắp xếp theo tên

/Sap xep theo tên

```
char ten_i[30],ten_j[30];
for(i = 0; i < N - 1; i ++){
    for(j = i +1; j < N; j ++){
        strcpy(ten_i,strchr(DS[i],32));
        strcpy(ten_j,strchr(DS[j],32));
        if(strcmp(ten_i,ten_j) > 0){
            strcpy(str,DS[i]);
            strcpy(DS[i],DS[j]);
            strcpy(DS[j],str);
        }
    }
}
```

So sinh vien : 4

Ten sinh vien[0]: NGUYEN VAN C

Ten sinh vien[1]: LE VAN A

Ten sinh vien[2]: TRAN THI D

Ten sinh vien[3]: LE THI F

DS sinh vien vua nhap

LE VAN A

NGUYEN VAN C

TRAN THI D

LE THI F



# Bài Tập Thao Tác Trên Chuỗi

# Bài 1 – Kiểm tra tính đối xứng

- Hãy viết chương trình cho phép:
  - Nhập xâu s từ bàn phím. Kiểm tra tính đối xứng của xâu vừa nhập.
- Gợi ý: Giả sử chuỗi đối xứng, thực hiện giải thuật bác bỏ. Nếu có 1 cặp ký tự  $(i, n - i - 1)$  khác nhau  $\Rightarrow$  chuỗi không đối xứng. Ở đây n là độ dài của chuỗi.

## Bài 2 – Tần suất xuất hiện của 1 ký tự

- Hãy viết chương trình cho phép:
  - Nhập vào 1 chuỗi và 1 ký tự, kiểm tra ký tự có trong chuỗi hay không, nếu có đưa ra số lần xuất hiện của ký tự đó trong chuỗi.

## Bài 3 – Ký tự viết hoa/thường

- Viết chương trình chuyển chuỗi nhập từ bàn phím thành:
  - Chữ viết hoa,
  - Viết thường và
  - Viết hoa mỗi chữ cái đầu mỗi từ.

# Bài 4 – Danh sách sinh viên

- Viết chương trình cho phép:
  - Nhập vào 1 danh sách sinh viên và hiển thị danh sách sinh viên ra màn hình

Memory location(base address)	Array elements									
25860	t	r	e	e	\0					
25870	b	o	w	l	\0					
25880	h	a	t	\0						
25890	m	i	c	e	\0					
25900	t	o	o	n	\0					

[5] names stored in 5 different memory locations

**Mảng chuỗi 2 chiều**

length of each String is [10]

## Bài 5 – Đảo ngược chuỗi (Không dùng string.h)

- Hãy viết một chương trình cho phép đảo ngược chuỗi nhập vào từ bàn phím.
  - Yêu cầu: Không được sử dụng các hàm có sẵn của string.h
- Ví dụ về input và output của chương trình

```
Nhap chuoì: Nguyen Van Hieu
```

```
Chuoi sau khi dao nguoc la: ueiH naV neyugN
```

```
Chuoi sau khi dao nguoc la: ueiH naV neyugN
```

## Bài 5 – Đảo ngược chuỗi (dùng string.h)

- Hãy viết một chương trình cho phép đảo ngược chuỗi nhập vào từ bàn phím.
  - Gợi ý: Ngôn ngữ C có thư viện `string.h`, trong đó có hàm `strrev()` giúp chúng ta có thể đảo ngược chuỗi dễ dàng
- Ví dụ về input và output của chương trình

```
Nhap chuoì: Nguyen Van Hieu
```

```
Chuoi sau khi dao nguoc la: ueiH naV neyugN
```

```
Chuoi sau khi dao nguoc la: ueiH naV neyugN
```



# Q&A

