



TIN ĐẠI CƯƠNG

Chủ đề: Giải quyết vấn đề trên máy tính bằng lập trình



Giới thiệu môn học

- Thời lượng: 3 tín chỉ (2/3 lý thuyết)
- Giáo trình chính
 - Introduction to Engineering Programming: Solving Problems with Algorithms (James Paul Holloway)
 - Đã có bản dịch tiếng Việt
- Công cụ trên máy tính: Dev-C++ 5.11
 - Hoặc những công cụ tương đương



Giới thiệu môn học

- Giới thiệu chung
- Các lệnh cơ bản
- Câu lệnh lựa chọn
- Câu lệnh lặp
- Kiểu dữ liệu và làm việc với dữ liệu
- Mạng
- Các kiểu kết hợp



Giới thiệu môn học

- Tính điểm:
 - Điểm bài tập (20%)
 - Điểm kiểm tra giữa kì (20%)
 - Điểm kiểm tra cuối kì (60%, thi viết)
- Giảng viên:
 - Tên: Trương Xuân Nam
 - Email: truongxuannam@gmail.com



Một vài chú ý khác

- Cần xem giáo trình trước khi lên lớp
- Phải làm hết bài tập (trong giáo trình và bài tập giao thêm)
- Yêu cầu hỗ trợ của giáo viên khi cần thiết
- Cố gắng đọc tiếng Anh
- Tài liệu được đưa lên <http://txnam.net>



Bài 1: Giới thiệu chung

- Mở đầu
- Thuật toán
- Các cấu trúc điều khiển
- Máy tính và lập trình cho máy tính
- Giới thiệu ngôn ngữ C/C++



Bài 1: Giới thiệu chung

- Mở đầu
- Thuật toán
- Các cấu trúc điều khiển
- Máy tính và lập trình cho máy tính
- Giới thiệu ngôn ngữ C/C++



1.1 Mở đầu

- Lý do phải học
 - Làm quen với máy tính theo cách của giới làm kỹ thuật
 - Hiểu cách thức giải quyết một vấn đề bằng máy tính
 - Nâng cao tư duy logic và tư duy thuật toán
 - Lấy kiến thức
 - Lấy bằng đại học



1.1 Mở đầu

- Mục tiêu của môn học
 - Cơ bản về ngôn ngữ lập trình C/C++
 - Cách triển khai một số thuật toán trên máy tính
 - Cách viết, dịch, sửa lỗi và chạy một chương trình viết bằng C++
 - Giải được một số bài toán đơn giản bằng lập trình C++



Bài 1: Giới thiệu chung

- Mở đầu
- Thuật toán
- Các cấu trúc điều khiển
- Máy tính và lập trình cho máy tính
- Giới thiệu ngôn ngữ C/C++



1.2 Thuật toán

- Định nghĩa: Các bước cần tiến hành để giải quyết một công việc cụ thể nào đó
- Đặc trưng:
 - Tính hữu hạn
 - Tính máy móc
 - Tính dừng
 - Mở rộng: Tính đúng
 - Mở rộng: Tính tổng quát



1.2 Thuật toán

Ví dụ (trong giáo trình): Tính bình phương của số m

Bước 1: Nhập giá trị cho m

Bước 2: Tính giá trị $m \times m$ và đưa vào s

Bước 3: Trả về giá trị s cho chương trình gọi



1.2 Thuật toán

Ví dụ: Giải phương trình $ax^2 + bx + c = 0$ với điều kiện $a \neq 0$

- Bước 1: Nhập các giá trị a, b, c
- Bước 2: Nếu $a = 0$ thì thông báo lỗi và dừng
- Bước 3: Tính $d = b^2 - 4 \times a \times c$
- Bước 4: Nếu $d < 0$ thì thông báo vô nghiệm và dừng
- Bước 5: Nếu $d > 0$ thì thực hiện Bước 7
- Bước 6: Thông báo có nghiệm $x = -b/2/a$ và dừng
- Bước 7: Thông báo có hai nghiệm $x_1 = (-b + \sqrt{d})/2/a$ và $x_2 = (-b - \sqrt{d})/2/a$



1.2 Thuật toán

Ví dụ (vui): Cho con sư tử vào tủ lạnh

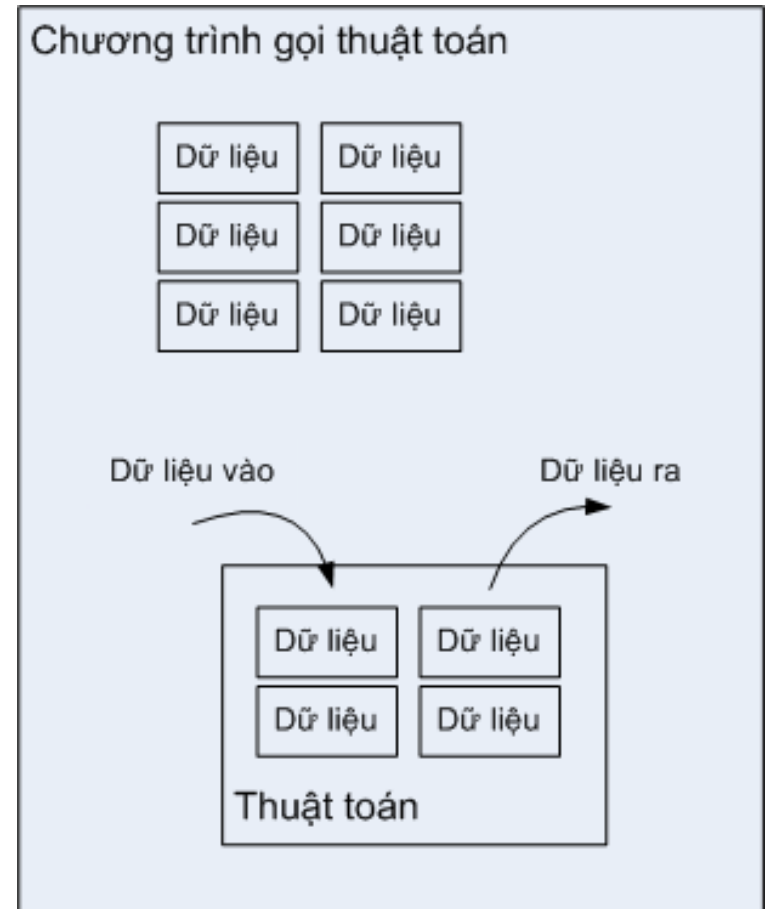
Bước 1: Mở cửa tủ lạnh

Bước 2: Cho con sư tử vào

Bước 3: Đóng cửa tủ lạnh

1.2.1 Môi trường thực thi

- Cung cấp dữ liệu để thuật toán hoạt động
- Cung cấp nơi nhận kết quả của thuật toán (dữ liệu ra)
- Mở rộng: Cung cấp ngữ cảnh để thuật toán hoạt động





1.2.1 Môi trường thực thi

- Như vậy thêm một đặc trưng mới cho Thuật toán: có giao diện (interface)
 - **Dữ liệu đầu vào** để thực hiện thuật toán
 - **Dữ liệu đầu ra** để thuật toán trả kết quả về cho môi trường thực thi
- Khái niệm
 - Truyền tham trị (pass-by-value)
 - Truyền tham chiếu (pass-by-reference)



Bài 1: Giới thiệu chung

- Mở đầu
- Thuật toán
- Các cấu trúc điều khiển
- Máy tính và lập trình cho máy tính
- Giới thiệu ngôn ngữ C/C++

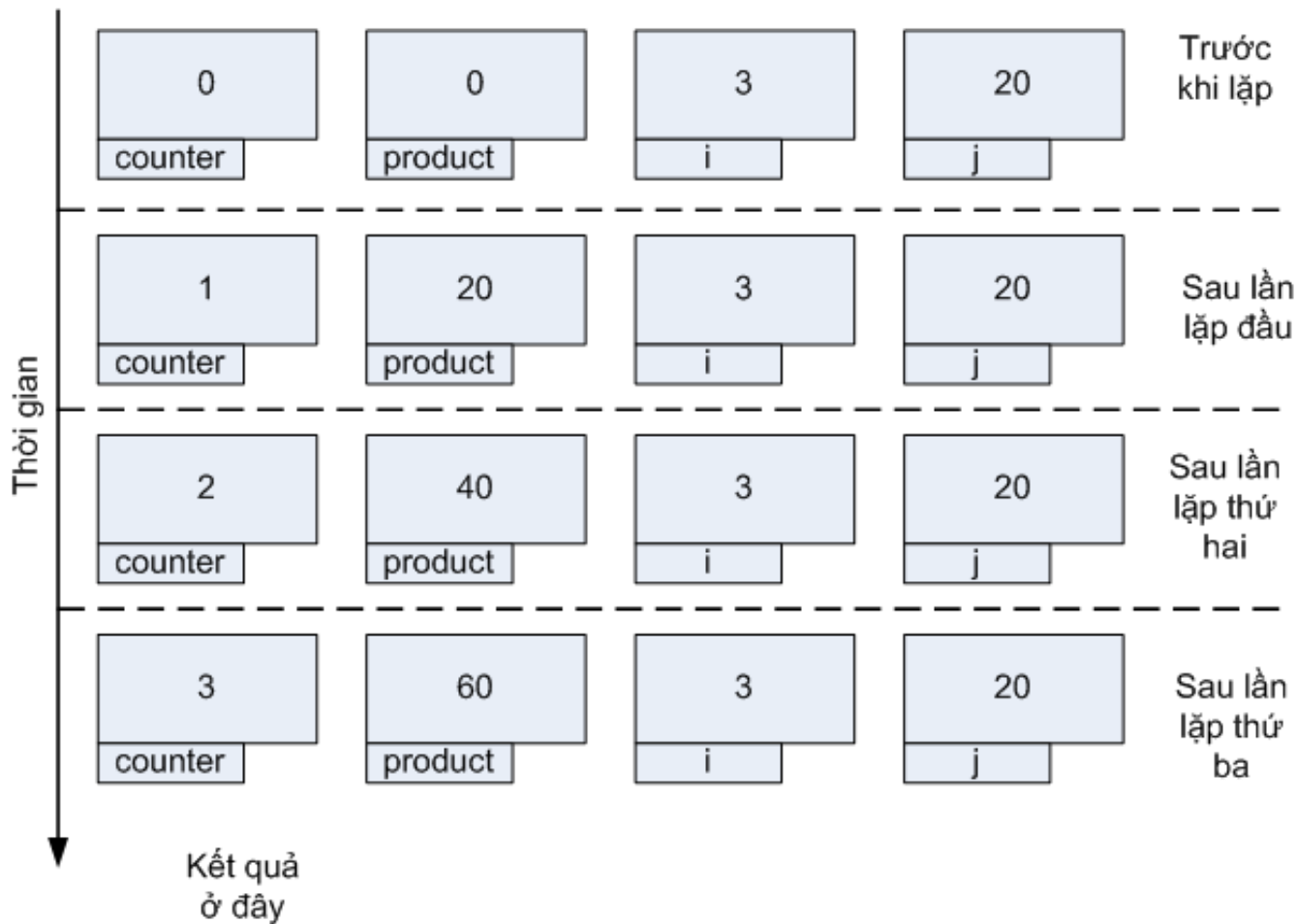


1.3 Các cấu trúc điều khiển

Xét ví dụ 1: Nhân 2 số tự nhiên i và j trên máy tính không có phép nhân

- 1: Đặt biến product về 0
- 2: Đặt biến counter về 0
- 3: **while** counter < i **do**
- 4: Đặt product bằng chính nó cộng với j
- 5: Tăng biến counter lên 1
- 6: **end while**
- 7: **return** giá trị của product

1.3 Các cấu trúc điều khiển





1.3 Các cấu trúc điều khiển

Ví dụ 2: Nhân 2 số nguyên i và j trên máy tính không có phép nhân

- **if** cả i và j đều không âm **then**
 - sử dụng thuật toán nhân không âm i và j và lưu kết quả trong `product`
- **else if** cả i và j đều âm **then**
 - đổi dấu cả i và j và lúc này chúng đều không âm
 - sử dụng thuật toán nhân không âm i và j và lưu kết quả trong `product`
- **else**



1.3 Các cấu trúc điều khiển

- **if** $i < 0$ **then**
 - đổi dấu của i
- **else**
 - đổi dấu của j
- **end if**
- sử dụng thuật toán nhân không âm i và j và lưu kết quả trong `product`
- đảo dấu của `product`
- **end if**
- **return** `product`



1.3 Các cấu trúc điều khiển

- **Tuần tự**: Thực hiện tuyến tính từng việc một
- **Lặp**: Thực hiện lặp lại một hoặc nhiều việc cho đến khi điều kiện nhất định được thỏa mãn
- **Lựa chọn (rẽ nhánh)**: Chọn thực hiện một hoặc nhiều việc dựa trên một điều kiện nhất định



Bài 1: Giới thiệu chung

- Mở đầu
- Thuật toán
- Các cấu trúc điều khiển
- Máy tính và lập trình cho máy tính
- Giới thiệu ngôn ngữ C/C++

1.4 Máy tính và lập trình cho máy tính

- Mọi thông tin đều có thể chuyển về dạng số:
 - Các số → giữ nguyên → số
 - Âm thanh → số hóa (tần số) → số
 - Hình ảnh → số hóa (ma trận điểm) → số
 - ...
- ➔ Máy tính xử lý các thông tin ở dạng số
- ➔ Mọi thông tin trong máy tính đều được lưu ở dạng số, cụ thể là số ở dạng nhị phân
- ➔ Máy tính chỉ hiểu các thông tin ở dạng số
- ➔ Ra lệnh cho máy tính làm việc phải **viết lệnh ở dạng số**

1.4 Máy tính và lập trình cho máy tính

- Máy tính chỉ hiểu một số lệnh cơ bản (lệnh máy):
 - Thao tác bộ nhớ: Ghi số vào ô nhớ / Đọc số từ ô nhớ ra CPU
 - Tính toán: Cộng 2 số, trừ 2 số,...
 - So sánh: So sánh 2 số với nhau
 - ...
- Chương trình máy tính = dãy các lệnh máy để chỉ thị cho máy tính là một việc cụ thể nào đó
- Kích thước một chương trình máy tính
 - Loại cực nhỏ: Vài trăm lệnh máy
 - Loại nhỏ: Vài chục nghìn lệnh máy
 - Loại vừa: Vài trăm nghìn lệnh máy
 - Loại lớn: Vài triệu lệnh máy

1.4 Máy tính và lập trình cho máy tính



- Bước 1: Người dùng ra lệnh cho máy tính thực hiện một chương trình
- Bước 2: Máy tính đọc file chương trình trên đĩa và nạp chương trình vào bộ nhớ
- Bước 3: Hệ thống có một số thao tác chuẩn bị để chương trình sẵn sàng chạy
- Bước 4: Máy tính đọc từng lệnh trong bộ nhớ và thực hiện từng lệnh một

1.4 Máy tính và lập trình cho máy tính

- Chương trình máy tính được ghi trên đĩa ở dạng file chương trình (.COM, .EXE, .DLL,...)
- Máy tính đọc lệnh máy trong bộ nhớ và thực hiện từng lệnh một:

00011000 00010000

00011001 00001111

00101010 10001001

Nạp số 16 vào ô nhớ số 8

Nạp số 15 vào ô nhớ số 9

Cộng hai số ở ô nhớ số 8 và ô nhớ số 9 sau đó ghi kết quả vào ô nhớ số 10

1.4 Máy tính và lập trình cho máy tính

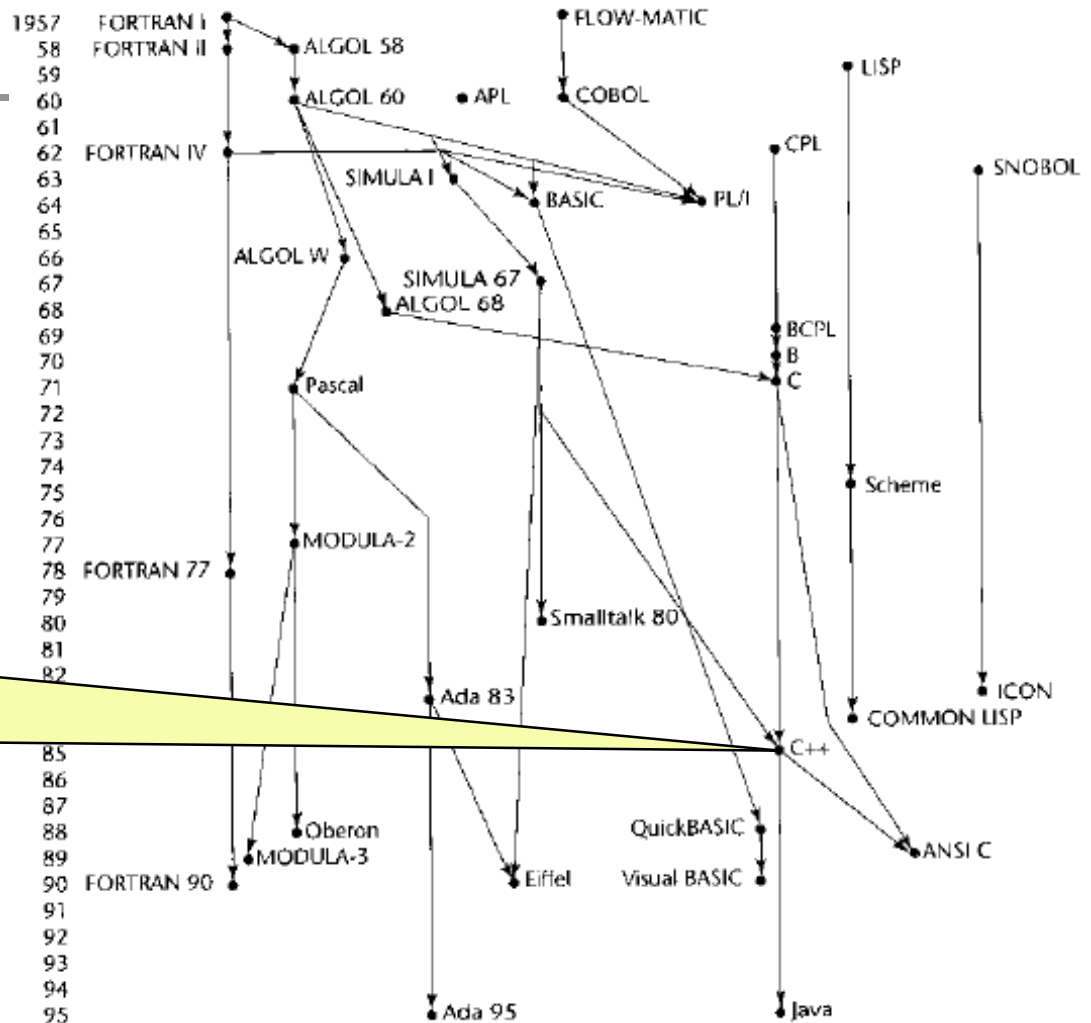
- Thời kì đầu: Viết trực tiếp lệnh máy
 - Bất lợi: Khó hiểu, dễ nhầm lẫn, viết lâu,...
- Hợp ngữ: Sử dụng các kí hiệu đơn giản bằng tiếng Anh, gắn gũi với lệnh máy
 - Bất lợi: Người lập trình phải biết rõ về từng lệnh máy



1.4 Máy tính và lập trình cho máy tính

- **Ngôn ngữ lập trình bậc cao:** Các lệnh được viết ở dạng gần gũi với ngôn ngữ tự nhiên, trình biên dịch chuyển một lệnh này thành các lệnh máy
- Ngôn ngữ lập trình bậc cao chia thành nhiều loại:
 - Ngôn ngữ bậc cao đơn giản: BASIC, FORTRAN,...
 - Ngôn ngữ lập trình thủ tục: ALGOL, PASCAL, C,...
 - Ngôn ngữ lập trình hướng đối tượng: SmallTalk, C++, Object Pascal, Java, C#,...
- **Các ngôn ngữ lập trình đặc biệt:** Prolog, SQL,...

1.4 Máy tính và lập trình cho máy tính



Ngôn ngữ lập trình C/C++

1.4 Máy tính và lập trình cho máy tính

- **Ngôn ngữ lập trình bậc cao:** Các lệnh được viết ở dạng gần gũi với ngôn ngữ tự nhiên, trình biên dịch chuyển một lệnh này thành các lệnh máy
- Ngôn ngữ lập trình bậc cao chia thành nhiều loại:
 - Ngôn ngữ bậc cao đơn giản: BASIC, FORTRAN,...
 - Ngôn ngữ lập trình thủ tục: ALGOL, PASCAL, C,...
 - Ngôn ngữ lập trình hướng đối tượng: SmallTalk, C++, Object Pascal, Java, C#,...
- **Các ngôn ngữ lập trình đặc biệt:** Prolog, SQL,...

1.4 Máy tính và lập trình cho máy tính

- Một chương trình máy tính được xây dựng để giải quyết một bài toán cụ thể nào đó.
- Việc xây dựng một chương trình máy tính luôn tuân theo các bước sau:
 - Bước 1: Xác định (mô tả) bài toán cần giải quyết
 - Bước 2: Xây dựng lời giải (thuật toán)
 - Bước 3: Chuyển lời giải bài toán thành chương trình viết bằng một ngôn ngữ lập trình nào đó
 - Bước 4: Dịch chương trình thành dạng mã máy để máy tính có thể thực hiện được

1.4 Máy tính và lập trình cho máy tính

- Bước 1 - xác định (mô tả) bài toán cần giải quyết:
 - Ví dụ: Bài toán tính A^2
 - Xác định bài toán: Người dùng cho số A , máy tính cần tính A^2 dựa trên số A đã biết
- Bước 2 - xây dựng lời giải (thuật toán):
 - Có nhiều cách mô tả thuật toán (bằng lời hoặc bằng sơ đồ khối)
 - Ví dụ (mô tả bằng lời): Nhập A từ bàn phím, sau đó tính giá trị $A \times A$ và in kết quả ra màn hình.

1.4 Máy tính và lập trình cho máy tính

- Bước 3 - chuyển lời giải bài toán thành chương trình viết bằng một ngôn ngữ lập trình nào đó:
 - Chọn ngôn ngữ lập trình thích hợp với bài toán
 - Viết chương trình theo thuật toán đã định
- Bước 4 - dịch chương trình thành dạng mã máy để máy tính có thể thực hiện được:
 - Sử dụng trình biên dịch của ngôn ngữ đã chọn và dịch chương trình sang dạng mã máy
 - Nếu xảy ra lỗi, tìm và sửa lỗi trong chương trình sau đó dịch lại đến khi không còn lỗi nữa



Bài 1: Giới thiệu chung

- Mở đầu
- Thuật toán
- Các cấu trúc điều khiển
- Máy tính và lập trình cho máy tính
- Giới thiệu ngôn ngữ C/C++

1.5 Giới thiệu ngôn ngữ C/C++



- Công cụ Dev-C++
- Hướng dẫn cơ bản:
 - Viết chương trình
 - Dịch
 - Chạy
 - Sửa lỗi
- Một vài ví dụ đơn giản