Bumbii Academy

GIÁO TRÌNH LẬP TRÌNH

C++ từ cơ bản đến nâng cao

17 chương lý thuyết có ví dụ chạy được · 102 bài tập
Bộ test chấm tự động trên Themis

Bản xem thử · 2 chương mẫu

tong.cpp

```cpp
#include <iostream>
using namespace std;

int main() {
    int n;
    cin >> n;
    long long sum = 0;
    for (int i = 1; i <= n; i++) {
        sum += i;          // cộng dồn từ 1 tới n
    }
    cout << sum << '\n';
    return 0;
}
```

Biến và kiểu dữ liệu · Rẽ nhánh · Vòng lặp · Hàm · Mảng · Xâu ký tự
vector, sort · struct · Đệ quy · set, map, stack, queue · Con trỏ
Hướng đối tượng · Cộng dồn · Hai con trỏ · Tìm kiếm nhị phân

bumbii.tech · youtube.com/@bumbii.academy · facebook.com/bumbii.academy

Tài liệu được biên soạn bởi ©2026 Bumbii Academy (https://bumbii.tech). Đây là **bản xem thử**, được chia sẻ tự do với điều kiện giữ nguyên nội dung và nguồn.

Mọi ý kiến đóng góp hoặc câu hỏi, xin vui lòng liên hệ tác giả:

Th.S Trần Huy Phúc

Kỹ sư phần mềm @Adnovum

Sáng lập Bumbii Academy & Bumbii English

Email: phuc@bumbii.tech

Zalo: 0972465159

Facebook: facebook.com/tran.huyphuc.3

Bumbii Academy

Về bản xem thử này

Đây là **bản xem thử** của bộ *Giáo trình C++ từ cơ bản đến nâng cao* do Bumbii Academy biên soạn, để thầy cô xem cách trình bày, cách giải thích và chất lượng bài tập trước khi quyết định dùng bộ tài liệu đầy đủ. Bản xem thử được chia sẻ tự do.

Bản xem thử gồm **trọn vẹn 2 chương**, giữ nguyên như trong bản đầy đủ:

- **Chương 1**: chương mở đầu, cho học sinh mới học lập trình.
- **Chương 8**: một chương ở giữa giáo trình, để thầy cô thấy độ sâu của phần lý thuyết và bài tập khi kiến thức đã nhiều hơn.
- **Lời giải chi tiết** 12 bài tập của 2 chương trên, đúng như bản có lời giải.
- **Danh sách 17 chương và 102 bài tập** của bản đầy đủ ở cuối tài liệu.

Mỗi chương gồm **mục tiêu**, **lý thuyết** (mỗi ý có chương trình ví dụ chạy được, kèm kết quả in ra, đã biên dịch thử với chuẩn C++17) và **6 bài tập** chỉ dùng kiến thức tới chương đó. Bài tập đọc dữ liệu từ bàn phím (`cin`), in kết quả ra màn hình (`cout`), có ví dụ lấy từ bộ test. Lời giải gồm ý tưởng, các bước giải, lỗi hay gặp và code C++ hoàn chỉnh. Bấm "Xem lời giải" ở mỗi bài để tới lời giải, bấm "Xem đề" để quay lại.

> **Bộ tài liệu đầy đủ** gồm: 2 file PDF (17 chương, 102 bài tập; một bản chỉ có đề để phát cho học sinh, một bản có lời giải), bài giảng PowerPoint cho từng chương (có ghi chú cho người dạy), bộ test chấm tự động trên Themis cho mọi bài tập (10 test mỗi bài, cấu hình sẵn) và code lời giải. Thông tin liên hệ ở trang cuối.

Bumbii Academy

Mục lục

Chương 1. Làm quen với C++ 4

Bài tập chương 1 12

Chương 8. Mảng một chiều 16

Bài tập chương 8 24

Lời giải bài tập 27

Lời giải chương 1. Làm quen với C++ 27

Lời giải chương 8. Mảng một chiều 31

Bộ giáo trình đầy đủ 37

Bumbii Academy

CHƯƠNG 1

Làm quen với C++

Mục tiêu

- Biết C++ là gì và quy trình viết, biên dịch, chạy một chương trình
- Hiểu cấu trúc của một chương trình C++ đơn giản
- In được chữ, số và kết quả phép tính ra màn hình bằng cout
- Dùng được ký tự đặc biệt (xuống dòng, tab, dấu ngoặc kép) và chú thích
- Đọc hiểu được thông báo lỗi biên dịch thường gặp

C++ là gì?

C++ là một **ngôn ngữ lập trình** do Bjarne Stroustrup phát triển từ năm 1979, dựa trên ngôn ngữ C. Đến nay C++ vẫn được dùng rộng rãi vì chương trình viết bằng C++ chạy **rất nhanh**: game, trình duyệt web, hệ điều hành, phần mềm nhúng đều có phần lớn mã nguồn bằng C++.

Với học sinh, C++ là ngôn ngữ được dùng nhiều trong các kỳ thi học sinh giỏi Tin học và Olympic Tin học, vì tốc độ chạy nhanh và thư viện chuẩn (STL) có sẵn nhiều cấu trúc dữ liệu, thuật toán.

Máy tính không hiểu trực tiếp mã C++. Chương trình phải đi qua ba bước:

Bước	Việc làm	Kết quả
1. Viết mã	Gõ mã nguồn vào trình soạn thảo	File `.cpp` (mã nguồn)
2. Biên dịch	Trình biên dịch (compiler) kiểm tra và dịch mã nguồn sang mã máy	File chạy được (trên Windows có đuôi `.exe`)
3. Chạy	Hệ điều hành chạy file kết quả	Chương trình làm việc, in kết quả

Nếu mã nguồn viết sai quy tắc của ngôn ngữ, bước 2 báo **lỗi biên dịch** và không tạo ra file chạy được. Ta phải sửa mã rồi biên dịch lại.

Bumbii Academy

Chuẩn bị công cụ

Để lập trình C++ cần hai thứ: **trình biên dịch** (phổ biến nhất là GCC, có tên lệnh `g++`) và **trình soạn thảo** để gõ mã. Các môi trường lập trình (IDE) dưới đây gom cả hai vào một:

- **Code::Blocks** (bản có chữ `mingw` trong tên file cài đặt): nhẹ, miễn phí, hay dùng ở trường học.
- **Dev-C++** (bản Embarcadero): rất nhẹ, phù hợp máy cấu hình thấp.
- **Visual Studio Code** + phần mở rộng C/C++ + trình biên dịch MinGW: hiện đại, cần cài đặt nhiều bước hơn.

Khi thi học sinh giỏi, thường dùng chuẩn **C++17** hoặc mới hơn. Trong Code::Blocks, bật tùy chọn này ở *Settings → Compiler → Have g++ follow the C++17 ISO C++ language standard*.

Nếu dùng dòng lệnh, biên dịch file `hello.cpp` bằng lệnh:

```
g++ -std=c++17 -o hello hello.cpp
```

Phần `-o hello` đặt **tên** cho file chạy được; **đuôi** file do hệ điều hành quyết định:

Hệ điều hành	Có `-o hello`	Không có `-o`	Lệnh chạy
Windows	`hello.exe` (tự thêm đuôi)	`a.exe`	`hello`
macOS, Linux	`hello` (không có đuôi)	`a.out`	`./hello`

Trong PowerShell của Windows, lệnh chạy là `.\hello`. Các IDE như Code::Blocks, Dev-C++ tự đặt file chạy được trùng tên với file mã nguồn (`hello.exe`) và tự chạy khi bấm Run.

Chương trình đầu tiên

```cpp
#include <iostream>

int main() {
    std::cout << "Xin chao C++!";
    return 0;
}
```

Kết quả trên màn hình:

```
Xin chao C++!
```

Bumbii Academy

Ý nghĩa từng dòng:

Dòng	Ý nghĩa
`#include <iostream>`	Nạp thư viện `iostream` (input/output stream), thư viện chứa `cout` để in ra màn hình.
`int main() { ... }`	Hàm `main` là **điểm bắt đầu** của mọi chương trình C++. Máy chạy lần lượt các lệnh bên trong cặp ngoặc `{ }` từ trên xuống.
`std::cout << "..."`	In nội dung trong dấu ngoặc kép ra màn hình. `std::` cho biết `cout` thuộc thư viện chuẩn.
`;`	Dấu chấm phẩy **kết thúc mỗi câu lệnh**. Thiếu dấu này là lỗi hay gặp nhất của người mới học.
`return 0;`	Kết thúc `main` và báo cho hệ điều hành rằng chương trình chạy thành công.

Viết gọn với `using namespace std;`

Gõ `std::` trước mỗi `cout` khá dài. Dòng `using namespace std;` cho phép bỏ phần này:

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Xin chao C++!";
    return 0;
}
```

Trong các chương trình nhỏ và bài thi, cách viết này rất phổ biến; tài liệu này cũng dùng cách viết đó. Trong dự án lớn, người ta thường giữ `std::` để tránh trùng tên.

> **Quy tắc viết mã dễ đọc.** Mỗi câu lệnh một dòng. Các lệnh bên trong `{ }` lùi vào 4 dấu cách (hoặc một phím Tab). Trình biên dịch không bắt buộc điều này, nhưng mã lùi đầu dòng đúng giúp đọc và tìm lỗi nhanh hơn nhiều.

Bumbii Academy

In dữ liệu với `cout`

Toán tử `<<` "đẩy" dữ liệu ra màn hình. Có thể nối nhiều `<<` trong một câu lệnh:

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Lop " << 8 << "A" << '\n';
    cout << "Si so: " << 35 << " hoc sinh" << '\n';
    return 0;
}
```

```
Lop 8A
Si so: 35 hoc sinh
```

Lưu ý ba loại dữ liệu khác nhau trong ví dụ trên:

- `"Lop "` là một **xâu ký tự**, đặt trong dấu ngoặc kép `" "`. Mọi ký tự bên trong (kể cả dấu cách) được in nguyên văn.
- `8`, `35` là **số**, không cần dấu ngoặc kép.
- `'\n'` là **một ký tự** đặt trong dấu ngoặc đơn `' '`. Ở đây là ký tự xuống dòng.

Xuống dòng: `'\n'` và `endl`

`cout` **không tự xuống dòng**. Muốn xuống dòng phải in ký tự `'\n'` hoặc `endl`:

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Dong 1";
    cout << "Dong 2" << endl;   // endl: xuống dòng
    cout << "Dong 3\n";         // \n viết ngay trong xâu
    cout << "Dong 4" << '\n';
    return 0;
}
```

```
Dong 1Dong 2
Dong 3
Dong 4
```

"Dong 1" và "Dong 2" dính liền nhau vì sau "Dong 1" không có lệnh xuống dòng. Hai cách `endl` và `'\n'` cho cùng kết quả trên màn hình; khác biệt là `endl` còn ép đẩy dữ liệu ra ngay (flush) nên chậm hơn khi in rất nhiều dòng. Vì vậy khi thi, nên dùng `'\n'`.

Bumbii Academy

In kết quả phép tính

`cout` in được giá trị của biểu thức. Máy **tính trước rồi mới in**:

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << 12 + 30 << '\n';           // 42
    cout << "12 + 30" << '\n';         // 12 + 30   (xâu: in nguyên văn)
    cout << "12 + 30 = " << 12 + 30 << '\n';
    cout << 7 * 8 << ' ' << 100 - 1 << '\n';
    return 0;
}
```

```
42
12 + 30
12 + 30 = 42
56 99
```

Dòng thứ hai cho thấy sự khác biệt quan trọng: khi nằm trong dấu ngoặc kép, `12 + 30` chỉ là chữ, không được tính. Phép nhân trong C++ viết là `*`, phép chia là `/` (chương 3 sẽ học kỹ).

Ký tự đặc biệt (escape sequence)

Một số ký tự không gõ trực tiếp vào xâu được, ví dụ dấu ngoặc kép (vì nó đánh dấu kết thúc xâu). Ta viết chúng bằng dấu gạch chéo ngược `\` đi kèm một ký tự:

Viết	Ý nghĩa
`\n`	Xuống dòng
`\t`	Tab (nhảy tới cột kế tiếp, thường cách 8 ký tự)
`\"`	Dấu ngoặc kép `"`
`\'`	Dấu nháy đơn `'`
`\\`	Dấu gạch chéo ngược `\`

Bumbii Academy

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Thay noi: \"Hoc C++ that vui!\"\n";
    cout << "Duong dan: C:\\BaiTap\\bai1.cpp\n";
    cout << "Ten\tTuoi\n";
    cout << "An\t13\n";
    return 0;
}
```

```
Thay noi: "Hoc C++ that vui!"
Duong dan: C:\BaiTap\bai1.cpp
Ten     Tuoi
An      13
```

Về tiếng Việt có dấu. Màn hình dòng lệnh trên Windows có thể hiển thị sai chữ có dấu. Vì vậy các bài tập trong tài liệu in kết quả **không dấu**. Chú thích trong mã nguồn thì vẫn viết có dấu bình thường.

Chú thích (comment)

Chú thích là phần ghi chú cho người đọc mã, trình biên dịch **bỏ qua** hoàn toàn.

```cpp
#include <iostream>
using namespace std;

/* Chương trình in thông tin lớp học.
   Chú thích nhiều dòng nằm giữa dấu mở và dấu đóng. */
int main() {
    // Chú thích một dòng: từ hai dấu gạch chéo tới hết dòng
    cout << "Lop 8A\n";     // chú thích có thể đặt cuối dòng lệnh
    // cout << "Dong nay khong chay\n";
    return 0;
}
```

Hai cách dùng chú thích hay gặp:

1. **Giải thích ý tưởng** của đoạn mã (vì sao làm thế), không lặp lại điều mã đã nói rõ.
2. **Tạm tắt một dòng lệnh** khi tìm lỗi: thêm `//` ở đầu dòng thay vì xóa dòng đó.

Bumbii Academy

Lỗi thường gặp và cách đọc thông báo lỗi

Có ba loại lỗi:

Loại lỗi	Khi nào phát hiện	Ví dụ
Lỗi biên dịch (cú pháp)	Lúc biên dịch, trình biên dịch báo dòng lỗi	Thiếu `;`, viết sai tên `cuot`, thiếu dấu `"`
Lỗi khi chạy (runtime)	Lúc chạy, chương trình dừng đột ngột	Chia cho 0, truy cập ngoài mảng (học sau)
Lỗi logic	Chương trình chạy nhưng kết quả sai	Viết `7 + 8` thay vì `7 * 8`

Ví dụ một chương trình có lỗi biên dịch:

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Xin chao"
    cout << "C++\n";
    return 0;
}
```

Trình biên dịch g++ báo:

```
hello.cpp:5:23: error: expected ';' before 'cout'
```

Cách đọc: `hello.cpp:5:23` là **file : dòng : cột**; `expected ';'` nghĩa là "thiếu dấu `;`". Lỗi thật nằm ở cuối dòng 5, vì trình biên dịch chỉ phát hiện ra khi gặp dòng 6. Kinh nghiệm: **khi báo lỗi ở một dòng mà dòng đó đúng, hãy xem dòng ngay phía trên**.

Một số lỗi khác:

- `'cout' was not declared in this scope`: quên `#include <iostream>` hoặc quên `using namespace std;`, hoặc gõ sai chính tả.
- `missing terminating " character`: thiếu dấu ngoặc kép đóng xâu.
- `expected '}' at end of input`: thiếu dấu ngoặc nhọn đóng.

Khi gặp nhiều lỗi cùng lúc, **sửa lỗi đầu tiên trước rồi biên dịch lại**: một lỗi thường kéo theo nhiều thông báo lỗi phía sau.

Bumbii Academy

Tóm tắt

- Chương trình C++: viết mã nguồn `.cpp` → biên dịch → chạy.
- Khung chương trình: `#include <iostream>`, `using namespace std;`, hàm `int main() { ... }`.
- `cout << ...` in ra màn hình; nối nhiều giá trị bằng nhiều `<<`.
- Xâu trong `" "` in nguyên văn; biểu thức ngoài ngoặc kép được tính rồi mới in.
- Xuống dòng bằng `'\n'` (nhanh) hoặc `endl`.
- Ký tự đặc biệt: `\n`, `\t`, `\"`, `\\`.
- Chú thích: `// một dòng`, `/* nhiều dòng */`.
- Mỗi câu lệnh kết thúc bằng `;`. Gặp lỗi: đọc số dòng, xem cả dòng phía trên, sửa lỗi đầu tiên trước.

Bumbii Academy

Bài tập chương 1

Bài 1.1. Xin chào

Mã bài: `XINCHAO` · Thời gian: 1 giây · Xem lời giải

Viết chương trình in ra màn hình đúng hai dòng sau:

```
Xin chao C++!
Toi bat dau hoc lap trinh.
```

Dữ liệu vào: Không có.

Kết quả: Hai dòng như trên.

Ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)
(không có)	`Xin chao C++!` `Toi bat dau hoc lap trinh.`

Bài 1.2. Hình chữ nhật

Mã bài: `HINHCN` · Thời gian: 1 giây · Xem lời giải

Viết chương trình in ra hình chữ nhật gồm 4 dòng, mỗi dòng 6 dấu `*` viết liền nhau.

Dữ liệu vào: Không có.

Kết quả: 4 dòng, mỗi dòng là `******`.

Ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)
(không có)	`******` `******` `******` `******`

Bumbii Academy

Bài 1.3. Tam giác sao

Mã bài: `TAMGIAC` · Thời gian: 1 giây · Xem lời giải

Viết chương trình in ra tam giác vuông gồm 5 dòng: dòng thứ nhất có 1 dấu `*`, dòng thứ hai có 2 dấu `*`, ..., dòng thứ năm có 5 dấu `*`. Các dấu `*` trên cùng một dòng viết liền nhau.

Dữ liệu vào: Không có.

Kết quả: 5 dòng như mô tả.

Ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)
(không có)	* ** *** **** *****

Bài 1.4. Bảng nhân 7

Mã bài: `BANGNHAN` · Thời gian: 1 giây · Xem lời giải

Viết chương trình in bảng nhân 7, gồm 10 dòng từ `7 x 1 = 7` đến `7 x 10 = 70`.

Yêu cầu: kết quả của mỗi phép nhân phải do **máy tính toán** bằng phép nhân `*` trong `cout`, không tự tính rồi gõ sẵn con số.

Dữ liệu vào: Không có.

Kết quả: 10 dòng, dòng thứ i có dạng `7 x i = <kết quả>`.

Ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)
(không có)	7 x 1 = 7 7 x 2 = 14 7 x 3 = 21 7 x 4 = 28 7 x 5 = 35 7 x 6 = 42 7 x 7 = 49 7 x 8 = 56 7 x 9 = 63 7 x 10 = 70

Bumbii Academy

Bài 1.5. Ký tự đặc biệt

Mã bài: `KYTUDB` · Thời gian: 1 giây · Xem lời giải

Viết chương trình in ra đúng ba dòng sau (chú ý các dấu ngoặc kép và dấu gạch chéo ngược):

```
Thay giao noi: "Lap trinh la nghe thuat."
File bai lam: D:\HSG\bai1.cpp
Ky tu \n dung de xuong dong.
```

Dữ liệu vào: Không có.

Kết quả: Ba dòng như trên.

Ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)

(không có)	`Thay giao noi: "Lap trinh la nghe thuat."`
	`File bai lam: D:\HSG\bai1.cpp`
	`Ky tu \n dung de xuong dong.`

Bài 1.6. Tấm thiệp

Mã bài: `THIEP` · Thời gian: 1 giây · Xem lời giải

Viết chương trình in ra tấm thiệp chúc mừng có viền như sau:

```
+----------------------+
|                      |
|  CHUC MUNG NAM MOI   |
|        2027          |
|                      |
+----------------------+
```

Dữ liệu vào: Không có.

Kết quả: 6 dòng như trên. Bài được chấm theo từng từ trên mỗi dòng nên số dấu cách giữa các từ không ảnh hưởng tới kết quả, nhưng các bạn nên căn chỉnh cho đẹp như mẫu.

Bumbii Academy

Ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)
(không có)	+----------------------+ | | | CHUC MUNG NAM MOI | | 2027 | | | +----------------------+

Bumbii Academy

CHƯƠNG 8

Mảng một chiều

Mục tiêu

- Khai báo mảng, truy cập phần tử bằng chỉ số, nhập và in mảng
- Duyệt mảng để tính tổng, tìm lớn nhất, đếm, tìm kiếm
- Đảo ngược, chèn, xóa phần tử trong mảng
- Dùng mảng đếm (mảng đánh dấu) để thống kê giá trị
- Sắp xếp mảng bằng thuật toán chọn và bằng hàm sort

Vì sao cần mảng?

Nhập điểm của 40 học sinh rồi in ra những bạn có điểm cao hơn điểm trung bình. Muốn biết điểm trung bình phải đọc hết 40 điểm, rồi lại cần **xem lại từng điểm** để so sánh. Khai báo 40 biến `score1`, `score2`, ... là không thể. **Mảng** là một dãy biến cùng kiểu, nằm liền nhau trong bộ nhớ, gọi chung một tên và phân biệt bằng **chỉ số**.

Khai báo và truy cập

```cpp
#include <iostream>
using namespace std;

int main() {
    int scores[5] = {7, 9, 4, 10, 6};   // mảng 5 số nguyên, khởi tạo sẵn
    cout << scores[0] << '\n';          // 7: phần tử đầu tiên có chỉ số 0
    cout << scores[4] << '\n';          // 6: phần tử cuối có chỉ số 4 = 5 - 1
    scores[2] = 8;                      // gán lại phần tử thứ ba
    for (int i = 0; i < 5; i++) {
        cout << scores[i] << ' ';       // 7 9 8 10 6
    }
    cout << '\n';
    return 0;
}
```

Chỉ số	0	1	2	3	4
`scores[i]`	7	9	8	10	6

- Mảng n phần tử có chỉ số từ **0 đến n − 1**.

Bumbii Academy

- `int a[5] = {};` khởi tạo mọi phần tử bằng 0. `int a[5] = {1, 2};` cho 1, 2, 0, 0, 0.
- Kích thước mảng khai báo bằng một **hằng số**.

Nhập và in mảng có n phần tử

Đề thường cho n trước, rồi n số. Ta khai báo mảng **đủ lớn cho n lớn nhất** theo giới hạn đề:

```cpp
#include <iostream>
using namespace std;

const int MAX_N = 100005;      // n ≤ 10^5, thêm vài phần tử dự phòng
long long a[MAX_N];            // mảng lớn khai báo toàn cục

int main() {
    int n;
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    for (int i = n - 1; i >= 0; i--) {
        cout << a[i] << ' ';   // in theo thứ tự ngược
    }
    cout << '\n';
    return 0;
}
```

Vì sao khai báo mảng lớn ngoài hàm main? Biến cục bộ nằm trong vùng nhớ "stack", thường chỉ có vài MB; mảng 10^6 phần tử khai báo trong `main` có thể làm chương trình dừng đột ngột. Mảng toàn cục nằm ở vùng nhớ khác, rộng hơn nhiều, và được tự động khởi tạo bằng 0.

Truy cập ngoài mảng. C++ không kiểm tra chỉ số: với mảng 5 phần tử, `a[5]` hay `a[-1]` vẫn biên dịch được nhưng đọc/ghi vào vùng nhớ của biến khác. Chương trình có thể chạy sai âm thầm hoặc dừng đột ngột. Luôn kiểm tra chỉ số nằm trong khoảng 0..n − 1.

Bumbii Academy

Duyệt mảng

Các mẫu đã học ở chương vòng lặp áp dụng y nguyên, chỉ khác là dữ liệu đã nằm sẵn trong mảng nên **duyệt được nhiều lần**:

```cpp
#include <iostream>
using namespace std;

int main() {
    int n = 6;
    int a[] = {4, -1, 9, 3, 9, 2};      // không ghi kích thước: tự lấy theo số phần tử
    long long sum = 0;
    int maxPos = 0;                     // vị trí của phần tử lớn nhất (đầu tiên)
    for (int i = 0; i < n; i++) {
        sum += a[i];
        if (a[i] > a[maxPos]) maxPos = i;
    }
    // Lần duyệt thứ hai: đếm số phần tử lớn hơn trung bình cộng
    int aboveAverage = 0;
    for (int i = 0; i < n; i++) {
        if ((long long)a[i] * n > sum) aboveAverage++;   // a[i] > sum / n, không dùng số thực
    }
    cout << sum << ' ' << a[maxPos] << ' ' << maxPos << ' ' << aboveAverage << '\n';   // 26 9 2 2
    return 0;
}
```

Mẹo so sánh với trung bình cộng: `a[i] > sum / n` tương đương `a[i] * n > sum` (n dương), tránh được phép chia số thực.

Bumbii Academy

Tìm kiếm tuyến tính

```cpp
#include <iostream>
using namespace std;

int main() {
    int a[] = {5, 8, 1, 8, 3};
    int n = 5, x = 8;
    int position = -1;                  // -1: chưa tìm thấy
    for (int i = 0; i < n; i++) {
        if (a[i] == x) {
            position = i;
            break;                      // lấy vị trí đầu tiên
        }
    }
    cout << position << '\n';           // 1
    return 0;
}
```

Vòng lặp for duyệt phần tử

C++11 có cách viết gọn khi chỉ cần **giá trị** từng phần tử, không cần chỉ số:

```cpp
#include <iostream>
using namespace std;

int main() {
    int a[] = {3, 1, 4, 1, 5};
    int sum = 0;
    for (int x : a) {       // x lần lượt là 3, 1, 4, 1, 5
        sum += x;
    }
    cout << sum << '\n';    // 14
    return 0;
}
```

Cách này duyệt **toàn bộ** mảng theo kích thước khai báo, nên chỉ dùng với mảng khai báo vừa đúng số phần tử (hoặc với `vector` ở chương 11).

Bumbii Academy

Biến đổi mảng

Đảo ngược một đoạn

Đổi chỗ phần tử đầu với cuối, rồi tiến hai đầu vào giữa:

```cpp
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
    int a[] = {1, 2, 3, 4, 5, 6, 7};
    int left = 1, right = 5;            // đảo đoạn a[1..5]
    while (left < right) {
        swap(a[left], a[right]);        // swap có sẵn trong thư viện
        left++;
        right--;
    }
    for (int x : a) cout << x << ' ';   // 1 6 5 4 3 2 7
    cout << '\n';
    return 0;
}
```

Chèn và xóa

Phần tử trong mảng nằm liền nhau, nên chèn hay xóa ở giữa phải **dời** các phần tử phía sau:

```cpp
#include <iostream>
using namespace std;

int main() {
    int a[10] = {10, 20, 30, 40};
    int n = 4;
    // Chèn 25 vào vị trí 2: dời a[2..n-1] sang phải một ô, đi từ cuối về
    int pos = 2, value = 25;
    for (int i = n; i > pos; i--) a[i] = a[i - 1];
    a[pos] = value;
    n++;                                    // 10 20 25 30 40
    // Xóa phần tử ở vị trí 0: dời các phần tử sau sang trái một ô
    for (int i = 0; i < n - 1; i++) a[i] = a[i + 1];
    n--;                                    // 20 25 30 40
    for (int i = 0; i < n; i++) cout << a[i] << ' ';
    cout << '\n';
    return 0;
}
```

Khi chèn phải dời **từ cuối về** (nếu dời từ đầu, phần tử sau bị ghi đè trước khi kịp dời). Mỗi lần chèn, xóa tốn khoảng n bước.

Bumbii Academy

Mảng đếm (mảng đánh dấu)

Khi giá trị các phần tử **nhỏ** (ví dụ từ 0 đến 10^6), dùng chính giá trị làm chỉ số của một mảng khác để đếm số lần xuất hiện:

```cpp
#include <iostream>
using namespace std;

const int MAX_VALUE = 1000000;
int countOf[MAX_VALUE + 1];             // toàn cục: tự khởi tạo bằng 0

int main() {
    int a[] = {3, 7, 3, 1, 7, 3};
    int n = 6;
    for (int i = 0; i < n; i++) {
        countOf[a[i]]++;                // giá trị a[i] xuất hiện thêm một lần
    }
    cout << countOf[3] << ' ' << countOf[7] << ' ' << countOf[5] << '\n';   // 3 2
        0
    return 0;
}
```

Sau một lần duyệt, ta trả lời ngay được "giá trị v xuất hiện bao nhiêu lần" bằng `countOf[v]`, không phải duyệt lại mảng. Kỹ thuật này dùng để: đếm tần suất, kiểm tra một giá trị đã xuất hiện chưa (mảng `bool seen[]`), đếm số giá trị khác nhau.

Bumbii Academy

Sắp xếp

Sắp xếp chọn (selection sort)

Ý tưởng: tìm phần tử nhỏ nhất, đưa về đầu; tìm phần tử nhỏ nhất trong phần còn lại, đưa về vị trí thứ hai; cứ thế tiếp tục.

```cpp
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
    int a[] = {5, 2, 9, 1, 6};
    int n = 5;
    for (int i = 0; i < n - 1; i++) {
        int minPos = i;
        for (int j = i + 1; j < n; j++) {
            if (a[j] < a[minPos]) minPos = j;
        }
        swap(a[i], a[minPos]);          // đưa phần tử nhỏ nhất của a[i..n-1] về vị trí i
    }
    for (int i = 0; i < n; i++) cout << a[i] << ' ';   // 1 2 5 6 9
    cout << '\n';
    return 0;
}
```

Lượt i	Mảng sau lượt
0	**1** 2 9 5 6
1	1 **2** 9 5 6
2	1 2 **5** 9 6
3	1 2 5 **6** 9

Hai vòng lặp lồng nhau cho khoảng $n^2/2$ phép so sánh: $n = 10^4$ cần 5×10^7 phép, vẫn kịp; $n = 10^5$ cần 5×10^9 phép, không kịp.

Bumbii Academy

Hàm sort của thư viện

Thư viện `<algorithm>` có hàm `sort` sắp xếp rất nhanh (khoảng $n \times \log_2 n$ phép so sánh, với $n = 10^6$ chỉ khoảng 2×10^7):

```cpp
#include <iostream>
#include <algorithm>
#include <functional>
using namespace std;

int main() {
    int a[] = {5, 2, 9, 1, 6};
    int n = 5;
    sort(a, a + n);                          // tăng dần: 1 2 5 6 9
    sort(a, a + n, greater<int>());          // giảm dần: 9 6 5 2 1
    for (int i = 0; i < n; i++) cout << a[i] << ' ';
    cout << '\n';
    return 0;
}
```

`sort(a, a + n)` sắp xếp các phần tử từ `a[0]` đến `a[n - 1]`. Muốn sắp xếp đoạn `a[l..r]` viết `sort(a + l, a + r + 1)`. Khi thi nên dùng `sort`; tự viết sắp xếp chọn để hiểu cách sắp xếp hoạt động.

Tóm tắt

- `int a[MAX_N];` chỉ số từ 0 đến n − 1; mảng lớn khai báo toàn cục với kích thước theo giới hạn đề.
- Không truy cập ngoài mảng: C++ không báo lỗi.
- Duyệt mảng nhiều lần được: tổng, lớn nhất, đếm, tìm kiếm.
- Đảo đoạn bằng hai chỉ số tiến vào giữa; chèn dời từ cuối, xóa dời từ đầu.
- Mảng đếm: `countOf[giá_trị]++` khi giá trị nhỏ.
- Sắp xếp chọn $O(n^2)$; `sort(a, a + n)` nhanh, có sẵn.

Bumbii Academy

Bài tập chương 8

Bài 8.1. Thống kê dãy số

Mã bài: `TONGMANG` · Thời gian: 1 giây · Xem lời giải

Cho dãy gồm n số nguyên $a_1, a_2, ..., a_n$. Hãy tính tổng của dãy, tìm giá trị lớn nhất, giá trị nhỏ nhất, và vị trí xuất hiện **đầu tiên** của giá trị lớn nhất (các vị trí đánh số từ 1).

Dữ liệu vào: Dòng đầu là số nguyên n ($1 \le n \le 10^5$). Dòng thứ hai gồm n số nguyên ($|a_i| \le 10^9$).

Kết quả: Một dòng gồm bốn số: tổng, giá trị lớn nhất, giá trị nhỏ nhất, vị trí đầu tiên của giá trị lớn nhất.

Ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)
6 4 -1 9 3 9 2	26 9 -1 3

Bài 8.2. Lớn hơn trung bình

Mã bài: `LONHONTB` · Thời gian: 1 giây · Xem lời giải

Cho dãy gồm n số nguyên. Hãy đếm số phần tử **lớn hơn** trung bình cộng của cả dãy.

Dữ liệu vào: Dòng đầu là số nguyên n ($1 \le n \le 10^5$). Dòng thứ hai gồm n số nguyên ($|a_i| \le 10^9$).

Kết quả: Số phần tử lớn hơn trung bình cộng.

Các ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)
6 4 -1 9 3 9 2	2
3 5 5 5	0

Giải thích ví dụ 1: trung bình cộng là 26 ÷ 6 ≈ 4,33; có 2 phần tử (9 và 9) lớn hơn.

Bumbii Academy

Bài 8.3. Đảo đoạn

Mã bài: `DAODOAN` · Thời gian: 1 giây · Xem lời giải

Cho dãy gồm n số nguyên và hai vị trí l, r (vị trí đánh số từ 1). Hãy đảo ngược thứ tự các phần tử từ vị trí l đến vị trí r, giữ nguyên các phần tử còn lại, rồi in dãy mới.

Dữ liệu vào: Dòng đầu gồm ba số nguyên n, l, r ($1 \le l \le r \le n \le 10^5$). Dòng thứ hai gồm n số nguyên ($|a_i| \le 10^9$).

Kết quả: Dãy sau khi đảo, trên một dòng, các số cách nhau một dấu cách.

Ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)
7 2 6 1 2 3 4 5 6 7	1 6 5 4 3 2 7

Bài 8.4. Sắp xếp

Mã bài: `SAPXEP` · Thời gian: 1 giây · Xem lời giải

Cho dãy gồm n số nguyên. Hãy sắp xếp dãy theo thứ tự tăng dần.

Yêu cầu: tự viết thuật toán sắp xếp (sắp xếp chọn hoặc sắp xếp nổi bọt), sau đó có thể so sánh với cách dùng hàm `sort` của thư viện.

Dữ liệu vào: Dòng đầu là số nguyên n ($1 \le n \le 5000$). Dòng thứ hai gồm n số nguyên ($|a_i| \le 10^9$).

Kết quả: Dãy đã sắp xếp tăng dần, trên một dòng.

Ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)
5 5 2 9 1 6	1 2 5 6 9

Bài 8.5. Tần suất

Mã bài: `TANSUAT` · Thời gian: 1 giây · Xem lời giải

Cho dãy gồm n số nguyên không âm. Hãy tìm giá trị xuất hiện nhiều lần nhất trong dãy và số lần xuất hiện của nó. Nếu có nhiều giá trị cùng xuất hiện nhiều lần nhất, chọn giá trị nhỏ nhất.

Bumbii Academy

Dữ liệu vào: Dòng đầu là số nguyên n ($1 \le n \le 10^5$). Dòng thứ hai gồm n số nguyên ($0 \le a_i \le 10^6$).

Kết quả: Một dòng gồm giá trị tìm được và số lần xuất hiện của nó.

Các ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)
6 3 7 3 1 7 3	3 3
4 9 2 9 2	2 2

Bài 8.6. Đoạn tăng dài nhất

Mã bài: `DOANTANG` · Thời gian: 1 giây · Xem lời giải

Cho dãy gồm n số nguyên. Một **đoạn** là một dãy các phần tử đứng liên tiếp nhau trong dãy. Đoạn **tăng** là đoạn mà mỗi phần tử lớn hơn hẳn phần tử đứng trước nó (đoạn chỉ có một phần tử cũng là đoạn tăng). Hãy tìm đoạn tăng dài nhất; nếu có nhiều đoạn dài nhất, chọn đoạn bắt đầu sớm nhất.

Dữ liệu vào: Dòng đầu là số nguyên n ($1 \le n \le 10^5$). Dòng thứ hai gồm n số nguyên ($|a_i| \le 10^9$).

Kết quả: Một dòng gồm độ dài của đoạn tăng dài nhất và vị trí bắt đầu của nó (vị trí đánh số từ 1).

Các ví dụ:

Dữ liệu vào (bàn phím)	Kết quả (màn hình)
8 5 7 2 1 4 6 9 3	4 4
4 4 4 4 4	1 1

Giải thích ví dụ 1: đoạn 1, 4, 6, 9 bắt đầu ở vị trí 4 dài 4 phần tử.

Bumbii Academy

Lời giải bài tập

Chương 1. Làm quen với C++

Bài 1.1. Xin chào

Mã bài: `XINCHAO` · Xem đề

Mỗi dòng cần in là một xâu, đặt trong dấu ngoặc kép. Sau mỗi xâu in thêm `\n` để xuống dòng; nếu thiếu, hai câu sẽ dính liền trên cùng một dòng và bài bị chấm sai.

Code C++ (XINCHAO.cpp)

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Xin chao C++!\n";
    cout << "Toi bat dau hoc lap trinh.\n";
    return 0;
}
```

Bài 1.2. Hình chữ nhật

Mã bài: `HINHCN` · Xem đề

Hình chữ nhật 4 dòng nên dùng 4 lệnh `cout`, mỗi lệnh in một xâu `******` rồi xuống dòng. Chương 5 (vòng lặp) sẽ cho cách in hình có kích thước bất kỳ mà không phải chép lại lệnh.

Code C++ (HINHCN.cpp)

```cpp
#include <iostream>
using namespace std;

int main() {
    // Mỗi lệnh in một dòng gồm 6 dấu * rồi xuống dòng
    cout << "******\n";
    cout << "******\n";
    cout << "******\n";
    cout << "******\n";
    return 0;
}
```

Bumbii Academy

Bài 1.3. Tam giác sao

Mã bài: `TAMGIAC` · Xem đề

Dòng thứ i có đúng i dấu `*`, nên năm lệnh in lần lượt các xâu dài 1, 2, 3, 4, 5 ký tự.

Lưu ý: không đặt dấu cách giữa các dấu `*`. Trình chấm tách mỗi dòng thành các từ; `* *` là hai từ, còn `**` là một từ, nên hai cách in này bị coi là khác nhau.

Code C++ (TAMGIAC.cpp)

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "*\n";
    cout << "**\n";
    cout << "***\n";
    cout << "****\n";
    cout << "*****\n";
    return 0;
}
```

Bài 1.4. Bảng nhân 7

Mã bài: `BANGNHAN` · Xem đề

Mỗi dòng gồm hai phần:

- phần chữ `"7 x 3 = "` đặt trong ngoặc kép, được in nguyên văn;
- phần biểu thức `7 * 3` đặt ngoài ngoặc kép, máy tính ra 21 rồi mới in.

Nếu viết cả `"7 x 3 = 7 * 3"` trong ngoặc kép, màn hình sẽ hiện nguyên văn chữ `7 * 3` chứ không phải 21.

Bumbii Academy

Code C++ (BANGNHAN.cpp)

```cpp
#include <iostream>
using namespace std;

int main() {
    // "7 x 1 = " là xâu (in nguyên văn), 7 * 1 là biểu thức (máy tính rồi in)
    cout << "7 x 1 = " << 7 * 1 << '\n';
    cout << "7 x 2 = " << 7 * 2 << '\n';
    cout << "7 x 3 = " << 7 * 3 << '\n';
    cout << "7 x 4 = " << 7 * 4 << '\n';
    cout << "7 x 5 = " << 7 * 5 << '\n';
    cout << "7 x 6 = " << 7 * 6 << '\n';
    cout << "7 x 7 = " << 7 * 7 << '\n';
    cout << "7 x 8 = " << 7 * 8 << '\n';
    cout << "7 x 9 = " << 7 * 9 << '\n';
    cout << "7 x 10 = " << 7 * 10 << '\n';
    return 0;
}
```

Bài 1.5. Ký tự đặc biệt

Mã bài: `KYTUDB` · Xem đề

Ba ký tự cần chú ý:

Muốn in	Viết trong xâu
`"`	`\"`
`\`	`\\`
hai ký tự `\` và `n`	`\\n`

Dòng thứ ba là bẫy của bài: viết `\n` trong xâu sẽ xuống dòng chứ không in ra chữ `\n`. Phải viết `\\n`: `\\` in ra dấu `\`, sau đó là chữ `n` bình thường.

Bumbii Academy

Code C++ (KYTUDB.cpp)

```cpp
#include <iostream>
using namespace std;

int main() {
    // \" in ra dấu ngoặc kép, \\ in ra một dấu gạch chéo ngược
    cout << "Thay giao noi: \"Lap trinh la nghe thuat.\"\n";
    cout << "File bai lam: D:\\HSG\\bai1.cpp\n";
    // Muốn in hai ký tự \ và n (không xuống dòng) thì viết \\n
    cout << "Ky tu \\n dung de xuong dong.\n";
    return 0;
}
```

Bài 1.6. Tấm thiệp

Mã bài: `THIEP` · Xem đề

Chép từng dòng của tấm thiệp vào một xâu. Để căn viền đẹp, đếm cho mọi dòng có cùng độ dài (24 ký tự): mỗi dòng giữa bắt đầu và kết thúc bằng `|`, phần bên trong luôn dài 22 ký tự.

Code C++ (THIEP.cpp)

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "+----------------------+\n";
    cout << "|                      |\n";
    cout << "|  CHUC MUNG NAM MOI   |\n";
    cout << "|        2027          |\n";
    cout << "|                      |\n";
    cout << "+----------------------+\n";
    return 0;
}
```

Bumbii Academy

Chương 8. Mảng một chiều

Bài 8.1. Thống kê dãy số

Mã bài: `TONGMANG` · Xem đề

Lưu dãy vào mảng với chỉ số từ 1 đến n (khớp cách đánh số của đề, không phải cộng trừ 1 khi in vị trí). Một lần duyệt cập nhật đồng thời tổng, vị trí lớn nhất, vị trí nhỏ nhất.

- Lưu **vị trí** `maxPos` thay vì giá trị: biết vị trí thì biết luôn giá trị `a[maxPos]`.
- So sánh bằng `>` (không phải `>=`): gặp giá trị bằng giá trị lớn nhất hiện tại thì không cập nhật, nên giữ được vị trí **đầu tiên**. Ví dụ dãy 4 −1 9 3 9 2 có hai số 9, kết quả là vị trí 3.
- Tổng tới $10^5 \times 10^9 = 10^{14}$: `long long`.

Code C++ (TONGMANG.cpp)

```cpp
#include <iostream>
using namespace std;

const int MAX_N = 100005;
long long a[MAX_N];

int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {     // dùng chỉ số 1..n cho khớp cách đánh số của đề
        cin >> a[i];
    }
    long long sum = 0;
    int maxPos = 1, minPos = 1;
    for (int i = 1; i <= n; i++) {
        sum += a[i];
        if (a[i] > a[maxPos]) maxPos = i;   // dấu > (không phải >=) giữ vị trí đầu tiên
        if (a[i] < a[minPos]) minPos = i;
    }
    cout << sum << ' ' << a[maxPos] << ' ' << a[minPos] << ' ' << maxPos << '\n';
    return 0;
}
```

Bumbii Academy

Bài 8.2. Lớn hơn trung bình

Mã bài: `LONHONTB` · Xem đề

Cần hai lần duyệt: lần đầu tính tổng (mới biết trung bình cộng), lần hai đếm. Vì vậy phải lưu dãy vào mảng.

Tránh số thực: điều kiện $a_i >$ tổng ÷ n tương đương $a_i \times n >$ tổng (nhân hai vế với $n > 0$).

Vế trái tới $10^9 \times 10^5 = 10^{14}$, dùng `long long`. Nếu tính trung bình bằng `double` rồi so sánh, với số lớn có thể sai do sai số làm tròn (ví dụ dãy toàn 10^9 chỉ có một số $10^9 - 1$).

Ví dụ 2: dãy 5 5 5 có trung bình 5, không phần tử nào **lớn hơn** 5, đáp số 0.

Code C++ (LONHONTB.cpp)

```cpp
#include <iostream>
using namespace std;

const int MAX_N = 100005;
long long a[MAX_N];

int main() {
    int n;
    cin >> n;
    long long sum = 0;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
        sum += a[i];
    }
    // a[i] > sum / n  tương đương  a[i] * n > sum  (n > 0): so sánh số nguyên, không sai số
    int count = 0;
    for (int i = 0; i < n; i++) {
        if (a[i] * n > sum) count++;
    }
    cout << count << '\n';
    return 0;
}
```

Bài 8.3. Đảo đoạn

Mã bài: `DAODOAN` · Xem đề

Dùng hai chỉ số `left = l`, `right = r`: đổi chỗ `a[left]` với `a[right]`, rồi `left++`, `right--`, dừng khi `left >= right` (đã gặp nhau ở giữa).

Bumbii Academy

Ví dụ đảo đoạn [2, 6] của dãy 1..7:

left	right	Dãy sau khi đổi
2	6	1 **6** 3 4 5 **2** 7
3	5	1 6 **5** 4 **3** 2 7
4	4	dừng

Mỗi phần tử trong đoạn được đổi chỗ đúng một lần, khoảng (r − l) ÷ 2 lần đổi. Thư viện có sẵn `reverse(a + l, a + r + 1)` làm đúng việc này.

Code C++ (DAODOAN.cpp)

```cpp
#include <iostream>
#include <algorithm>
using namespace std;

const int MAX_N = 100005;
int a[MAX_N];

int main() {
    int n, left, right;
    cin >> n >> left >> right;
    for (int i = 1; i <= n; i++) cin >> a[i];
    // Đổi chỗ hai đầu đoạn rồi tiến dần vào giữa
    while (left < right) {
        swap(a[left], a[right]);
        left++;
        right--;
    }
    for (int i = 1; i <= n; i++) cout << a[i] << ' ';
    cout << '\n';
    return 0;
}
```

Bài 8.4. Sắp xếp

Mã bài: `SAPXEP` · Xem đề

Sắp xếp chọn (xem lý thuyết): lượt thứ i tìm vị trí phần tử nhỏ nhất trong a[i..n−1] rồi đổi về vị trí i. Sau lượt i, các phần tử a[0..i] đã đúng vị trí cuối cùng.

Số phép so sánh khoảng $n^2 \div 2 = 1{,}25 \times 10^7$ với n = 5000, chạy nhanh. Với $n = 10^5$ thì cần 5×10^9 phép, không kịp; khi đó dùng `sort(a, a + n)`, chỉ cần khoảng $n \times \log_2 n \approx 1{,}7 \times 10^6$ phép so sánh.

Code C++ (SAPXEP.cpp)

```cpp
#include <iostream>
#include <algorithm>
using namespace std;

const int MAX_N = 5005;
int a[MAX_N];

int main() {
    int n;
    cin >> n;
    for (int i = 0; i < n; i++) cin >> a[i];
    // Sắp xếp chọn: lượt i đưa phần tử nhỏ nhất của a[i..n-1] về vị trí i
    for (int i = 0; i < n - 1; i++) {
        int minPos = i;
        for (int j = i + 1; j < n; j++) {
            if (a[j] < a[minPos]) minPos = j;
        }
        swap(a[i], a[minPos]);
    }
    for (int i = 0; i < n; i++) cout << a[i] << ' ';
    cout << '\n';
    return 0;
}
```

Bài 8.5. Tần suất

Mã bài: `TANSUAT` · Xem đề

Giá trị chỉ từ 0 đến 10^6 nên dùng **mảng đếm** `countOf[v]`: đọc giá trị x nào thì tăng `countOf[x]` thêm 1. Mảng 10^6 phần tử phải khai báo toàn cục.

Sau đó duyệt v từ 0 đến 10^6 để tìm v có `countOf[v]` lớn nhất. Vì duyệt từ nhỏ đến lớn và chỉ cập nhật khi **lớn hơn hẳn**, khi có nhiều giá trị cùng số lần xuất hiện thì giữ được giá trị nhỏ nhất. Ví dụ 2: 9 và 2 đều xuất hiện 2 lần, kết quả là 2.

Số phép tính khoảng $n + 10^6$, rất nhanh. So với cách đếm từng giá trị bằng vòng lặp lồng nhau (khoảng $n^2 = 10^{10}$ phép), mảng đếm nhanh hơn rất nhiều.

Bumbii Academy

Code C++ (TANSUAT.cpp)

```cpp
#include <iostream>
using namespace std;

const int MAX_VALUE = 1000000;
int countOf[MAX_VALUE + 1];      // countOf[v]: số lần giá trị v xuất hiện

int main() {
    int n;
    cin >> n;
    for (int i = 0; i < n; i++) {
        int x;
        cin >> x;
        countOf[x]++;
    }
    // Duyệt giá trị từ nhỏ đến lớn, chỉ cập nhật khi lớn hơn hẳn: tự chọn giá trị
        nhỏ nhất
    int bestValue = 0;
    for (int v = 1; v <= MAX_VALUE; v++) {
        if (countOf[v] > countOf[bestValue]) bestValue = v;
    }
    cout << bestValue << ' ' << countOf[bestValue] << '\n';
    return 0;
}
```

Bài 8.6. Đoạn tăng dài nhất

Mã bài: `DOANTANG` · Xem đề

Ý chính: đi từ trái sang phải, luôn biết độ dài đoạn tăng **kết thúc tại vị trí hiện tại**.

- Nếu `a[i] > a[i - 1]`: phần tử i nối tiếp được đoạn đang có, `length++`.
- Ngược lại: đoạn bị đứt, bắt đầu đoạn mới chỉ gồm phần tử i (`length = 1`, `start = i`).

Mỗi bước so `length` với kết quả tốt nhất; dùng `>` để giữ đoạn bắt đầu sớm nhất. Chỉ một lần duyệt, khoảng n bước.

Bumbii Academy

Dò ví dụ 1 (5 7 2 1 4 6 9 3):

i	a[i]	length	start	best
2	7	2	1	2 (từ 1)
3	2	1	3	2
4	1	1	4	2
5	4	2	4	2
6	6	3	4	3 (từ 4)
7	9	4	4	4 (từ 4)
8	3	1	8	4

Lưu ý: đề yêu cầu tăng **hẳn**, nên hai phần tử bằng nhau làm đứt đoạn (ví dụ 2: dãy 4 4 4 4 có đáp số 1, bắt đầu ở vị trí 1).

Code C++ (DOANTANG.cpp)

```cpp
#include <iostream>
using namespace std;

const int MAX_N = 100005;
int a[MAX_N];

int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) cin >> a[i];
    int bestLength = 1, bestStart = 1;
    int length = 1, start = 1;        // đoạn tăng đang xét kết thúc tại vị trí i
    for (int i = 2; i <= n; i++) {
        if (a[i] > a[i - 1]) {
            length++;                 // nối dài đoạn hiện tại
        } else {
            length = 1;               // bắt đầu đoạn mới tại i
            start = i;
        }
        if (length > bestLength) {    // dấu > giữ đoạn bắt đầu sớm nhất
            bestLength = length;
            bestStart = start;
        }
    }
    cout << bestLength << ' ' << bestStart << '\n';
    return 0;
}
```

Bumbii Academy

Bộ giáo trình đầy đủ

Danh sách chương và bài tập

Các dòng tô xanh là chương có trong bản xem thử này.

Ch.	Chủ đề	Bài tập
1	**Làm quen với C++**	**Xin chào (`XINCHAO`), Hình chữ nhật (`HINHCN`), Tam giác sao (`TAMGIAC`), Bảng nhân 7 (`BANGNHAN`), Ký tự đặc biệt (`KYTUDB`), Tấm thiệp (`THIEP`)**
2	Biến, kiểu dữ liệu và nhập xuất	Tổng hai số (`TONG2`), Hoán đổi (`HOANVI`), Hình chữ nhật (`HCN`), Trung bình cộng (`TBC3`), Mã ký tự (`KYTU`), Đổi nhiệt độ (`NHIETDO`)
3	Toán tử và biểu thức	Đổi giây (`DOIGIAY`), Số có ba chữ số (`CHUSO3`), Chia kẹo (`CHIAKEO`), Xe đưa đón (`LAMTRON`), Khoảng cách (`KHOANGCACH`), Đồng hồ (`DONGHO`)
4	Câu lệnh rẽ nhánh	Chẵn hay lẻ (`CHANLE`), Lớn nhất, nhỏ nhất (`MAX3`), Năm nhuận (`NAMNHUAN`), Số ngày trong tháng (`SONGAY`), Phân loại tam giác (`TAMGIAC`), Tiền điện (`TIENDIEN`)
5	Vòng lặp for	Tổng và đếm số chẵn (`TONGN`), Liệt kê ước (`UOCSO`), Giai thừa (`GIAITHUA`), Tam giác số (`TAMGIACSO`), Số Fibonacci (`FIBO`), Kiểm tra số nguyên tố (`NGUYENTO`)
6	Vòng lặp while, do-while, break và continue	Đếm chữ số (`DEMCHUSO`), Số đảo ngược (`DAOSO`), Dãy kết thúc bằng 0 (`TONGDEN0`), Ước chung và bội chung (`UCLN`), Dãy Collatz (`COLLATZ`), Đổi sang hệ nhị phân (`NHIPHAN`)
7	Hàm	Sắp xếp ba số (`SAPXEP3`), Tổng chữ số chia hết (`TONGCS`), Rút gọn phân số (`PHANSO`), Số nguyên tố sinh đôi (`SINHDOI`), Số Armstrong (`ARMSTRONG`), Lũy thừa nhanh (`LUYTHUA`)
8	**Mảng một chiều**	**Thống kê dãy số (`TONGMANG`), Lớn hơn trung bình (`LONHONTB`), Đảo đoạn (`DAODOAN`), Sắp xếp (`SAPXEP`), Tần suất (`TANSUAT`), Đoạn tăng dài nhất (`DOANTANG`)**
9	Mảng hai chiều	Tổng hàng, tổng cột (`TONGHANG`), Chuyển vị (`CHUYENVI`), Hai đường chéo (`DUONGCHEO`), Dò mìn (`DOMIN`), Nhân ma trận (`NHANMT`), Ma trận xoắn ốc (`XOANOC`)
10	Xâu ký tự	Phân loại ký tự (`DEMKYTU`), Xâu đối xứng (`DOIXUNG`), Chuẩn hóa họ tên (`CHUANHOA`), Đếm từ (`DEMTU`), Nén xâu (`NENXAU`), Cộng số lớn (`CONGLON`)
11	Vector và thư viện algorithm	Sắp xếp giảm dần (`SAPGIAM`), Tổng k số lớn nhất (`TONGKLON`), Các giá trị khác nhau (`KHACNHAU`), Chẵn trước, lẻ sau (`CHANLE2`), Trộn hai dãy (`TRONDAY`), Đếm tam giác (`BOBATG`)

Bumbii Academy

12	Struct và pair	Điểm gần gốc tọa độ (`GANNHAT`), Học sinh giỏi (`HSGIOI`), Bảng xếp hạng (`XEPHANG`), Giao hai hình chữ nhật (`HCNGIAO`), Cộng phân số (`CONGPS`), Lịch họp (`LICHHOP`)
13	Đệ quy	Tổng chữ số đệ quy (`TONGCSDQ`), Ước chung của dãy (`UCLNDQ`), Tháp Hà Nội (`THAPHN`), Xâu nhị phân (`XAUNP`), Tập con có tổng S (`TAPCON`), Hoán vị (`HOANVI`)
14	Các cấu trúc dữ liệu set, map, stack, queue	Hàng đợi (`HANGDOI`), Tần suất từ (`TANSUATTU`), Hợp và giao (`TAPHOP`), Dãy ngoặc đúng (`NGOAC`), Cặp có tổng K (`CAPTONG`), Nối dây (`NOIDAY`)
15	Con trỏ và bộ nhớ động	Duyệt mảng bằng con trỏ (`TONGCT`), Mảng động và tổng tiền tố (`MANGDONG`), Thống kê qua con trỏ (`THONGKE`), Xoay ma trận (`XOAYMT`), Đảo từng từ (`DAOXAUC`), Danh sách liên kết (`DSLK`)
16	Lập trình hướng đối tượng	Lớp hình chữ nhật (`HINHCN2`), Tài khoản ngân hàng (`TAIKHOAN`), Lớp phân số (`PHANSOLOP`), Lớp thời gian (`THOIGIAN`), Các loại hình (`HINHHOC`), Số nguyên lớn (`SOLON`)
17	Kỹ thuật lập trình nâng cao	Tổng đoạn (`TONGDOAN`), Đếm số nhỏ hơn hoặc bằng (`DEMNHO`), Số nguyên tố trong đoạn (`SANGNT`), Cộng trên đoạn (`CONGDOAN`), Đoạn con dài nhất (`DOANCON`), Cắt gỗ (`CATGO`)

Bộ tài liệu đầy đủ gồm

- **File PDF lý thuyết và bài tập** (không lời giải), dùng để phát cho học sinh.
- **File PDF có lời giải**: như bản trên, thêm phần lời giải ở cuối cho mọi bài tập.
- **Bài giảng PowerPoint** cho từng chương: slide tóm tắt có code minh họa, mỗi slide có ghi chú gợi ý cách trình bày cho người dạy, sửa được tự do.
- **Bộ test chấm tự động trên Themis**: 10 test mỗi bài, đã đặt sẵn điểm, thời gian và cách so sánh; kèm bài làm mẫu để thử chấm lần đầu và hướng dẫn chấm.
- **Code lời giải C++** của mọi bài, đã chạy đúng toàn bộ bộ test.

Giá bộ tài liệu đầy đủ: 250.000đ. Liên hệ mua: tác giả Th.S Trần Huy Phúc, Zalo 0972465159, Facebook facebook.com/tran.huyphuc.3.

Bumbii Academy