GIÁO TRÌNH LẬP TRÌNH C++

HÀM VÀ CON TRỎ

Biên soạn:

Lê Hà Phi

Pinocchio1412

Năm 2025

MỤC LỤC

Chương 6: 10 Bài tập Kết hợp Hàm và Con trỏ

PHẦN I: LÝ THUYẾT CƠ BẢN

CHƯƠNG 1: GIỚI THIỆU VỀ HÀM TRONG C++

1.1 Khái niệm Hàm

Hàm (Function) là một khối lệnh được đặt tên, có chức năng thực hiện một công việc cụ thể. Một hàm có thể được gọi nhiều lần từ các nơi khác nhau trong chương trình, giúp:

- Chia nhỏ chương trình thành các phần dễ quản lý
- Tái sử dụng code (code reusability)
- Làm cho chương trình dễ hiểu và bảo trì hơn
- Giảm lỗi lập trình

1.2 Cấu trúc của Hàm

```
kiểu_trả_về tên_hàm(danh_sách_tham_số) { // phần thân hàm // các lệnh thực hiện return giá_trị; // nếu kiểu trả về không phải void }
```

Các thành phần:

- **Kiểu trả về (Return Type):** int, double, char, void, v.v.
- **Tên hàm (Function Name):** Tên do người lập trình đặt
- **Danh sách tham số (Parameters):** Dữ liệu đầu vào cho hàm (có thể không có)
- **Phần thân (Body):** Các lệnh thực hiện công việc của hàm

1.3 Ví dụ Hàm Đơn Giản

```
// Hàm tính tổng hai số int tong(int a, int b) { return a + b; } // Hàm in lời chào void sayHello() { cout << "Xin chào!"; }
```

1.4 Gọi Hàm (Calling a Function)

```
int main() { int kq = tong(3, 5); // Gọi hàm tong cout << "Kết quả: " << kq; // In ra: Kết quả: 8 sayHello(); // Gọi hàm sayHello return 0; }
```

1.5 Truyền Tham Trị (Pass by Value)

> **Định nghĩa:** Khi truyền tham trị, hàm nhận một **bản sao** của giá trị. Thay đổi trong hàm không ảnh hưởng đến biến gốc.

```
void tăng(int x) { x++; // Chỉ tăng bản sao, không ảnh hưởng đến biến gốc } int main() { int a = 5; tăng(a); cout << a; // In ra: 5 (không thay đổi) }
```

1.6 Truyền Tham Chiếu (Pass by Reference)

> **Định nghĩa:** Khi truyền tham chiếu (dùng &), hàm nhận **địa chỉ thực** của biến. Thay đổi trong hàm sẽ ảnh hưởng đến biến gốc.

```
void tăng(int &x) { x++; // Tăng biến gốc } int main() { int a = 5; tăng(a); cout << a; // In ra: 6 (biến gốc thay đổi) }
```

1.7 Hàm Đệ Quy (Recursive Function)

> Hàm đệ quy là hàm gọi chính nó. Phải có điều kiện dừng để tránh vòng lặp vô tận.

```
// Hàm tính giai thừa đệ quy int giaithua(int n) { if (n == 0 || n == 1) return 1; // Điều kiện dừng return n * giaithua(n - 1); }
```

CHƯƠNG 2: GIỚI THIỆU VỀ CON TRỎ TRONG C++

2.1 Khái niệm Con trỏ

> **Con trỏ (Pointer)** là một biến lưu trữ **địa chỉ bộ nhớ** của một biến khác.

Tại sao cần con trỏ?

- Truyền tham chiếu đến hàm (thay đổi biến gốc)
- Quản lý bộ nhớ động (new, delete)
- Làm việc với mảng, chuỗi ký tự
- Tạo cấu trúc dữ liệu phức tạp (danh sách liên kết, cây, v.v.)

2.2 Khai báo Con trỏ

```
// Khai báo con trỏ kiểu int int *p; // Khai báo con trỏ kiểu double double *q; // Khai báo con trỏ kiểu char char *c;
```

2.3 Toán tử & và *

Toán tử	Tên	Ý nghĩa
&	Address-of	Lấy địa chỉ của biến
*	Dereference	Truy cập giá trị tại địa chỉ mà con trỏ chỉ tới

2.4 Ví dụ Cơ Bản

```
int main() { int a = 10; int *p = &a; // p lưu địa chỉ của a cout << "Giá trị a: " << a; // 10 cout << "Địa chỉ a: " << p; // 0x7fff... (địa chỉ bộ nhớ) cout << "Giá trị
```

```
tại p: " << *p; // 10 *p = 20; // Thay đổi giá trị tại địa chỉ cout << "Giá trị a sau: " << a; // 20 }
```

2.5 Con trỏ và Mảng

> Tên mảng chính là con trỏ trỏ tới phần tử đầu tiên của mảng.

```
int arr[5] = {1, 2, 3, 4, 5}; int *p = arr; // p trỏ tới phần tử đầu cout << *p; // 1 (phần tử thứ 0) cout << *(p + 1); // 2 (phần tử thứ 1) cout << *(p + 2); // 3 (phần tử thứ 2)
```

2.6 Cấp Phát Bộ Nhớ Động

> **new:** Cấp phát bộ nhớ động
>
> **delete:** Giải phóng bộ nhớ đã cấp phát

```
// Cấp phát bộ nhớ cho 1 biến int *p = new int; *p = 10; delete p; // Cấp phát bộ nhớ cho mảng int *arr = new int[5]; arr[0] = 1; delete[] arr; // Lưu ý: dùng delete[] cho mảng
```

2.7 Con trỏ Hàm (Function Pointer)

> Con trỏ hàm lưu địa chỉ của một hàm, cho phép gọi hàm qua con trỏ.

```
// Khai báo con trỏ hàm int (*pf)(int, int); // Gán địa chỉ hàm pf = &tong; // Gọi hàm qua con trỏ int kq = (*pf)(3, 5);
```

CHƯƠNG 3: KẾT HỢP HÀM VÀ CON TRỎ

3.1 Sử Dụng Con trỏ trong Hàm

Khi truyền con trỏ vào hàm, ta có thể thay đổi giá trị của biến gốc hoặc làm việc với mảng động.

3.2 Ví dụ 1: Hoán Đổi Hai Giá Trị

```
void hoánĐổi(int *x, int *y) { int temp = *x; *x = *y; *y = temp; } int main() { int a = 5, b = 10; hoánĐổi(&a, &b); cout << a << " " << b; // 10 5 }
```

3.3 Ví dụ 2: Xử Lý Mảng Động

```
void nhập(int *arr, int n) { for(int i = 0; i < n; i++) cin >> arr[i]; } void in(int *arr, int n) { for(int i = 0; i < n; i++) cout << arr[i] << " "; } int main() { int n; cin >> n; int *arr = new int[n]; nhập(arr, n); in(arr, n); delete[] arr; }
```

3.4 Hàm Trả Về Con trỏ

```
int* tạoMảng(int n) { int *arr = new int[n]; for(int i = 0; i < n; i++) arr[i] = i + 1; return arr; } int main() { int *myArr = tạoMảng(5); // Sử dụng myArr delete[] myArr; }
```

3.5 Mảng Con trỏ (Array of Pointers)

```
// Mảng 3 con trỏ int *arr[3]; // Cấp phát bộ nhớ arr[0] = new int; arr[1] = new int; arr[2] = new int; // Giải phóng bộ nhớ delete arr[0]; delete arr[1]; delete arr[2];
```

3.6 Con trỏ Tới Con trỏ (Pointer to Pointer)

```
int a = 10; int *p = &a; // Con trỏ tới a int **pp = &p; // Con trỏ tới con trỏ cout << **pp; // 10 (giá trị của a)
```

3.7 Bảng So Sánh: Tham Trị vs Tham Chiếu vs Con Trỏ

Khía cạnh	Tham Trị	Tham Chiếu	Con Trỏ
Thay đổi biến gốc	Không	Có	Có
Cấp phát bộ nhớ	Tự động	Tự động	Thủ công (new/delete)
Null được phép	Không	Không	Có
Dễ sử dụng	Dễ nhất	Dễ	Phức tạp hơn

PHẦN II: BÀI TẬP THỰC HÀNH

CHƯƠNG 4: 10 BÀI TẬP VỀ HÀM

📑 Bài 1: Tính Giai Thừa

Đề bài: Viết hàm tính giai thừa của số nguyên n (n! = 1×2×3×...×n)

📌 Ví dụ:

Input:	Output:
5	120

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int giaithua(int n) { int gt = 1;
for(int i = 1; i <= n; i++) gt *= i; return gt; } int main() { int n; cin >> n;
cout << giaithua(n); return 0; }
```

📝 Bài 2: Tính Tổng Từ 1 Đến n

Đề bài: Viết hàm tính tổng: 1 + 2 + 3 + ... + n

📌 Ví dụ:

Input:	Output:
5	15

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int tong(int n) { int s = 0; for(int i = 1; i <= n; i++) s += i; return s; } int main() { int n; cin >> n; cout << tong(n); return 0; }
```

📝 Bài 3: Kiểm Tra Số Nguyên Tố

Đề bài: Viết hàm kiểm tra xem n có phải là số nguyên tố hay không

📌 Ví dụ:

Input:	Output:
7	Yes

💻 Mã tham khảo:

```
#include <iostream> using namespace std; bool nguyenTo(int n) { if(n < 2) return false; for(int i = 2; i*i <= n; i++) if(n % i == 0) return false; return true; } int main() { int n; cin >> n; if(nguyenTo(n)) cout << "Yes"; else cout << "No"; return 0; }
```

📝 Bài 4: Tính Ước Chung Lớn Nhất (UCLN)

Đề bài: Viết hàm tính UCLN của hai số a và b

📌 Ví dụ:

Input:	Output:
12 8	4

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int UCLN(int a, int b) { while(b != 0) { int r = a % b; a = b; b = r; } return a; } int main() { int a, b; cin >> a >> b; cout << UCLN(a, b); return 0; }
```

📝 Bài 5: Tính Bội Chung Nhỏ Nhất (BCNN)

Đề bài: Viết hàm tính BCNN của hai số a và b (biết BCNN = a×b / UCLN)

📌 Ví dụ:

Input:	Output:
12 8	24

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int UCLN(int a, int b) { while(b != 0) { int r = a % b; a = b; b = r; } return a; } int BCNN(int a, int b) { return a * b / UCLN(a, b); } int main() { int a, b; cin >> a >> b; cout << BCNN(a, b); return 0; }
```

📝 Bài 6: Tính Số Fibonacci Thứ n

Đề bài: Viết hàm tính số Fibonacci thứ n (F(0)=0, F(1)=1, F(n)=F(n-1)+F(n-2))

📌 Ví dụ:

Input:	Output:
6	8

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int fibo(int n) { if(n <= 1) return n; return fibo(n - 1) + fibo(n - 2); } int main() { int n; cin >> n; cout << fibo(n); return 0; }
```

📝 Bài 7: Tính Tổng Các Chữ Số

Đề bài: Viết hàm tính tổng các chữ số của một số nguyên n

📌 Ví dụ:

Input:	Output:
12345	15

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int tongChuSo(int n) { int s = 0; while(n > 0) { s += n % 10; n /= 10; } return s; } int main() { int n; cin >> n; cout << tongChuSo(n); return 0; }
```

📝 Bài 8: Đảo Ngược Số

Đề bài: Viết hàm đảo ngược các chữ số của một số n

> 📌 **Ví dụ:**
>
> | Input: | Output: |
> | --- | --- |
> | 12345 | 54321 |

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int daoNguoc(int n) { int rev = 0; while(n > 0) { rev = rev * 10 + n % 10; n /= 10; } return rev; } int main() { int n; cin >> n; cout << daoNguoc(n); return 0; }
```

📝 Bài 9: Tính Chỉnh Hợp A(n,k)

Đề bài: Viết hàm tính chỉnh hợp chập k của n: A(n,k) = n! / (n-k)!

> 📌 **Ví dụ:**
>
> | Input: | Output: |
> | --- | --- |
> | 5 3 | 60 |

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int giaithua(int n) { int gt = 1; for(int i = 1; i <= n; i++) gt *= i; return gt; } int chinhHop(int n, int k) { return giaithua(n) / giaithua(n - k); } int main() { int n, k; cin >> n >> k; cout << chinhHop(n, k); return 0; }
```

📝 Bài 10: Tính Tổ Hợp C(n,k)

Đề bài: Viết hàm tính tổ hợp chập k của n: C(n,k) = n! / (k! × (n-k)!)

📌 Ví dụ:

Input:	Output:
5 3	10

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int giaithua(int n) { int gt = 1; for(int i = 1; i <= n; i++) gt *= i; return gt; } int toHop(int n, int k) { return giaithua(n) / (giaithua(k) * giaithua(n - k)); } int main() { int n, k; cin >> n >> k; cout << toHop(n, k); return 0; }
```

CHƯƠNG 5: 10 BÀI TẬP VỀ CON TRỎ

📝 Bài 11: In Giá Trị và Địa Chỉ

Đề bài: Khai báo biến a, sử dụng con trỏ để in ra giá trị và địa chỉ của a

📌 Ví dụ:

Input:	Output:
10	10

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int main() { int a = 10; int *p = &a;
cout << "Giá trị: " << *p << endl; cout << "Địa chỉ: " << p << endl; return 0; }
```

📝 Bài 12: Hoán Đổi Hai Số Bằng Con Trỏ

Đề bài: Nhập hai số x, y. Sử dụng con trỏ để hoán đổi giá trị của chúng

📌 Ví dụ:

Input:	Output:
5 10	10 5

💻 Mã tham khảo:

```
#include <iostream> using namespace std; void swapPtr(int *x, int *y) { int temp = *x; *x = *y; *y = temp; } int main() { int a, b; cin >> a >> b; swapPtr(&a, &b); cout << a << " " << b; return 0; }
```

📝 Bài 13: Tính Tổng Mảng Dùng Con Trỏ

Đề bài: Nhập n và mảng n phần tử. Tính tổng các phần tử bằng con trỏ

📌 Ví dụ:

Input:	Output:
5 1 2 3 4 5	15

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int tongMang(int *a, int n) { int s = 0; for(int i = 0; i < n; i++) s += *(a + i); return s; } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); cout << tongMang(arr, n); delete[] arr; return 0; }
```

📝 Bài 14: Tìm Phần Tử Lớn Nhất

Đề bài: Nhập n và mảng n phần tử. Tìm phần tử lớn nhất bằng con trỏ

📌 Ví dụ:

Input:	Output:
5 1 9 2 8 3	9

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int timMax(int *a, int n) { int max = *a; for(int i = 1; i < n; i++) if(*(a + i) > max) max = *(a + i); return max; } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); cout << timMax(arr, n); delete[] arr; return 0; }
```

📝 Bài 15: Tìm Phần Tử Nhỏ Nhất

Đề bài: Nhập n và mảng n phần tử. Tìm phần tử nhỏ nhất bằng con trỏ

📌 Ví dụ:

Input:	Output:
5 1 9 2 8 3	1

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int timMin(int *a, int n) { int min = *a; for(int i = 1; i < n; i++) if(*(a + i) < min) min = *(a + i); return min; } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); cout << timMin(arr, n); delete[] arr; return 0; }
```

📝 Bài 16: Đếm Số Chẵn Trong Mảng

Đề bài: Nhập n và mảng n phần tử. Đếm số lượng phần tử chẵn bằng con trỏ

📌 Ví dụ:

Input:	Output:
6 1 2 3 4 5 6	3

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int demChan(int *a, int n) { int dem = 0; for(int i = 0; i < n; i++) if(*(a + i) % 2 == 0) dem++; return dem; } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); cout << demChan(arr, n); delete[] arr; return 0; }
```

📝 Bài 17: Đảo Ngược Mảng

Đề bài: Nhập n và mảng n phần tử. Đảo ngược mảng bằng con trỏ

📌 Ví dụ:

Input:	Output:
5 1 2 3 4 5	5 4 3 2 1

💻 Mã tham khảo:

```
#include <iostream> using namespace std; void daoMang(int *a, int n) { for(int i = 0; i < n/2; i++) { int temp = *(a + i); *(a + i) = *(a + n - 1 - i); *(a + n - 1 - i) = temp; } } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); daoMang(arr, n); for(int i = 0; i < n; i++) cout << *(arr + i) << " "; delete[] arr; return 0; }
```

📝 Bài 18: Sắp Xếp Mảng Tăng Dần

Đề bài: Nhập n và mảng n phần tử. Sắp xếp tăng dần bằng con trỏ

📌 Ví dụ:

Input:

```
5
4 2 5 1 3
```

Output:

```
1 2 3 4 5
```

💻 Mã tham khảo:

```
#include <iostream> using namespace std; void sortTang(int *a, int n) { for(int i = 0; i < n - 1; i++) { for(int j = i + 1; j < n; j++) { if(*(a + i) > *(a + j)) { int temp = *(a + i); *(a + i) = *(a + j); *(a + j) = temp; } } } } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); sortTang(arr, n); for(int i = 0; i < n; i++) cout << *(arr + i) << " "; delete[] arr; return 0; }
```

📝 Bài 19: Sắp Xếp Mảng Giảm Dần

Đề bài: Nhập n và mảng n phần tử. Sắp xếp giảm dần bằng con trỏ

📌 Ví dụ:

Input:	Output:
5 4 2 5 1 3	5 4 3 2 1

💻 Mã tham khảo:

```
#include <iostream> using namespace std; void sortGiam(int *a, int n) { for(int i = 0; i < n - 1; i++) { for(int j = i + 1; j < n; j++) { if(*(a + i) < *(a + j)) { int temp = *(a + i); *(a + i) = *(a + j); *(a + j) = temp; } } } } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); sortGiam(arr, n); for(int i = 0; i < n; i++) cout << *(arr + i) << " "; delete[] arr; return 0; }
```

📝 Bài 20: Cấp Phát Mảng Động 2 Chiều

Đề bài: Tạo mảng động 2 chiều kích thước m×n, nhập dữ liệu và in ra

📌 Ví dụ:

Input:	Output:
2 3 1 2 3 4 5 6	1 2 3 4 5 6

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int main() { int m, n; cin >> m >> n; int **a = new int*[m]; for(int i = 0; i < m; i++) a[i] = new int[n]; for(int i = 0; i < m; i++) for(int j = 0; j < n; j++) cin >> a[i][j]; for(int i = 0; i < m; i++) { for(int j = 0; j < n; j++) cout << a[i][j] << " "; cout << endl; } for(int i = 0; i < m; i++) delete[] a[i]; delete[] a; return 0; }
```

CHƯƠNG 6: 10 BÀI TẬP KẾT HỢP HÀM VÀ CON TRỎ

📑 Bài 21: Hàm Nhập Mảng Bằng Con Trỏ

Đề bài: Viết hàm nhập n phần tử của mảng sử dụng con trỏ

📌 Ví dụ:

Input:	Output:
3 5 10 15	5 10 15

💻 Mã tham khảo:

```
#include <iostream> using namespace std; void nhapMang(int *a, int n) { for(int i = 0; i < n; i++) cin >> *(a + i); } void inMang(int *a, int n) { for(int i = 0; i < n; i++) cout << *(a + i) << " "; } int main() { int n; cin >> n; int *arr = new int[n]; nhapMang(arr, n); inMang(arr, n); delete[] arr; return 0; }
```

📝 Bài 22: Hàm In Mảng Bằng Con Trỏ

Đề bài: Viết hàm in n phần tử của mảng sử dụng con trỏ (chia hàng mỗi 5 phần tử)

📌 Ví dụ:

Input:

```
8
1 2 3 4 5 6 7 8
```

Output:

```
1 2 3 4 5
6 7 8
```

💻 Mã tham khảo:

```
#include <iostream> using namespace std; void inMang(int *a, int n) { for(int i = 0; i < n; i++) { cout << *(a + i) << " "; if((i + 1) % 5 == 0) cout << endl; } } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); inMang(arr, n); delete[] arr; return 0; }
```

📝 Bài 23: Hàm Tìm Giá Trị Lớn Nhất (Hàm + Con Trỏ)

Đề bài: Viết hàm tìm phần tử lớn nhất trong mảng, nhập mảng và in kết quả

📌 Ví dụ:

Input:

```
5
3 7 2 9 4
```

Output:

```
9
```

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int maxArr(int *a, int n) { int max = *a; for(int i = 1; i < n; i++) if(*(a + i) > max) max = *(a + i); return max; } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); cout << maxArr(arr, n); delete[] arr; return 0; }
```

📝 Bài 24: Tính Trung Bình Cộng Mảng

Đề bài: Viết hàm tính trung bình cộng các phần tử của mảng (kết quả là số thực)

📌 Ví dụ:

Input:	Output:
4 2 4 6 8	5

💻 Mã tham khảo:

```
#include <iostream> using namespace std; double tbMang(int *a, int n) { int s = 0; for(int i = 0; i < n; i++) s += *(a + i); return (double)s / n; } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); cout << tbMang(arr, n); delete[] arr; return 0; }
```

📝 Bài 25: Kiểm Tra Mảng Tăng Dần

Đề bài: Viết hàm kiểm tra xem mảng có tăng dần hay không

📌 Ví dụ:

Input:	Output:
5 1 2 3 4 5	Yes

💻 Mã tham khảo:

```
#include <iostream> using namespace std; bool tangDan(int *a, int n) { for(int i = 1; i < n; i++) if(*(a + i) < *(a + i - 1)) return false; return true; } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); if(tangDan(arr, n)) cout << "Yes"; else cout << "No"; delete[] arr; return 0; }
```

📝 Bài 26: Đếm Số Lần Xuất Hiện Của Phần Tử

Đề bài: Viết hàm đếm số lần xuất hiện của phần tử x trong mảng

📌 Ví dụ:

Input:	Output:
7 3 1 2 3 3 4 3 5	3

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int demX(int *a, int n, int x) { int dem = 0; for(int i = 0; i < n; i++) if(*(a + i) == x) dem++; return dem; } int main() { int n, x; cin >> n >> x; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); cout << demX(arr, n, x); delete[] arr; return 0; }
```

📝 Bài 27: Đảo Ngược Chuỗi Ký Tự

Đề bài: Viết hàm đảo ngược chuỗi ký tự bằng con trỏ

📌 Ví dụ:

Input:	Output:
hello	olleh

💻 Mã tham khảo:

```
#include <iostream> #include <string.h> using namespace std; void daoChuoi(char *s) { int n = strlen(s); for(int i = 0; i < n/2; i++) { char temp = *(s + i); *(s + i) = *(s + n - 1 - i); *(s + n - 1 - i) = temp; } } int main() { char s[100]; cin >> s; daoChuoi(s); cout << s; return 0; }
```

📝 Bài 28: Kiểm Tra Chuỗi Palindrome

Đề bài: Viết hàm kiểm tra xem chuỗi có phải Palindrome (đối xứng) hay không

📌 Ví dụ:

Input:	Output:
madam	Yes

💻 Mã tham khảo:

```
#include <iostream> #include <string.h> using namespace std; bool isPalindrome(char *s) { int l = 0, r = strlen(s) - 1; while(l < r) { if(*(s + l) != *(s + r)) return false; l++; r--; } return true; } int main() { char s[100]; cin >> s; if(isPalindrome(s)) cout << "Yes"; else cout << "No"; return 0; }
```

📝 Bài 29: Hoán Đổi Hai Phần Tử Trong Mảng

Đề bài: Viết hàm hoán đổi hai phần tử tại vị trí i và j trong mảng

📌 Ví dụ:

Input:	Output:
5 0 4 1 2 3 4 5	5 2 3 4 1

💻 Mã tham khảo:

```
#include <iostream> using namespace std; void hoanDoi(int *a, int i, int j) { int tmp = *(a + i); *(a + i) = *(a + j); *(a + j) = tmp; } int main() { int n, i, j; cin >> n >> i >> j; int *arr = new int[n]; for(int k = 0; k < n; k++) cin >> *(arr + k); hoanDoi(arr, i, j); for(int k = 0; k < n; k++) cout << *(arr + k) << " "; delete[] arr; return 0; }
```

📝 Bài 30: Tìm Phần Tử Chẵn Lớn Nhất

Đề bài: Viết hàm tìm phần tử chẵn lớn nhất trong mảng bằng con trỏ

📌 Ví dụ:

Input:	Output:
6 3 10 5 12 8 9	12

💻 Mã tham khảo:

```
#include <iostream> using namespace std; int maxChan(int *a, int n) { int max = -1; for(int i = 0; i < n; i++) { if(*(a + i) % 2 == 0 && *(a + i) > max) max = *(a + i); } return max; } int main() { int n; cin >> n; int *arr = new int[n]; for(int i = 0; i < n; i++) cin >> *(arr + i); cout << maxChan(arr, n); delete[] arr; return 0; }
```

KẾT LUẬN

Tóm Tắt Kiến Thức

Hàm (Function):

- Là khối lệnh được đặt tên, thực hiện công việc cụ thể
- Hỗ trợ tái sử dụng code và chia nhỏ chương trình
- Có thể truyền tham trị hoặc tham chiếu
- Có thể trả về giá trị hoặc không (void)

Con Trỏ (Pointer):

- Lưu trữ địa chỉ bộ nhớ của biến khác
- Dùng & để lấy địa chỉ, * để truy cập giá trị
- Cho phép cấp phát bộ nhớ động (new/delete)
- Dùng với mảng, chuỗi, và cấu trúc dữ liệu phức tạp

Kết Hợp Hàm + Con Trỏ:

- Truyền con trỏ vào hàm để thao tác trên dữ liệu gốc
- Xử lý mảng động một cách hiệu quả
- Tạo hàm tái sử dụng và linh hoạt

Lưu Ý Quan Trọng

> ⚠️ **Các lỗi thường gặp:**
>
> - Quên delete/delete[] sau khi cấp phát bộ nhớ → Memory Leak
> - Truy cập con trỏ null → Crash chương trình
> - Sử dụng * và & nhầm lẫn
> - Truyền con trỏ vào hàm mà quên & trong lệnh gọi

Tiếp Theo Học Gì?

Sau khi nắm vững hàm và con trỏ, bạn có thể tiếp tục học:

- **Struct & Class:** Tổ chức dữ liệu phức tạp
- **Danh sách liên kết:** Cấu trúc dữ liệu động
- **File I/O:** Đọc ghi file
- **OOP (Lập trình hướng đối tượng):** Tư duy lập trình hiện đại
- **STL (Standard Template Library):** Thư viện chuẩn C++

Giáo trình Lập trình C++ - Hàm và Con trỏ

Biên soạn: Lê Hà Phi / Pinocchio1412

Năm 2025