

BÀI THỰC TẬP SỐ 5

Con trỏ và Hàm trong C++

Bài thực tập này gồm các nội dung chính sau:

- Con trỏ
- Mảng động
- Hàm
- đệ quy

Câu 1: Hãy cho biết kích thước của biến con trỏ kiểu *byte* và kiểu *long*

```
# include <iostream.h>
void main()
{
    byte* a;
    long* b;
    cout<<sizeof(a)<<endl;
    cout<<sizeof(b)<<endl;
}
```

Câu 2: Cho đoạn chương trình sau:

```
float pay;
float *ptr_pay;
pay=2313.54;
ptr_pay = &pay;
```

Hãy cho biết giá trị của:

- a. pay
- b. *ptr_pay
- c. *pay
- d. &pay

Câu 3: Đọc để hiểu chương trình C++ sau đây:

```
#include<iostream.h>
void main()
{
    int a;
    int *aPtr;    // aPtr is a pointer to an integer

    a = 7;
    aPtr = &a;    //aPtr set to address of a
    cout << "The address of a is " << &a
        << "\nThe value of aPtr is " << aPtr;
```

```

        cout << "\n\nThe value of a is " << a
        << "\nThe value of *aPtr is " << *aPtr
        << endl;
    }

```

Chạy thử chương trình trên và giải thích kết quả của chương trình.

Câu 4: Cho một hàm như sau:

```

int square(int a)
{
    a = a*a;
    return a;
}

```

- Viết chương trình C++ nhập vào số nguyên x và gọi hàm *square* để tính bình phương của x và trình bày kết quả này ra.
- Viết lại hàm *square* để hàm này trở thành một hàm gọi bằng địa chỉ, đặt tên hàm mới là *square2*. Viết chương trình C++ nhập vào số nguyên x và gọi hàm *square2* để tính bình phương của x và trình bày giá trị của x sau khi gọi hàm. Có nhận xét gì về giá trị của x sau khi gọi hàm?

Câu 5: Đọc hàm sau đây dùng để tính số nguyên lớn nhất mà bình phương của nó nhỏ hơn hay bằng một trị số nguyên cho trước.

```

int Intqrt(int num)
{
    int i;
    i = 1;
    do
        ++ i
    while i*i <= num;
    return(i -1);
}

```

Viết một chương trình C++ đọc vào một số nguyên n và gọi hàm Intqrt để tính số nguyên lớn nhất mà bình phương của nó $\leq n$.

Câu 5: Hãy khai báo mảng động cho ma trận hai chiều có kích thước $m \times n$. Sau đó, kiểm tra có phải ma trận đối xứng không.

Câu 6: Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), sắp xếp các chữ số của n theo thứ tự tăng dần.

Ví dụ: Nhập $n = 1536$

Kết quả sau khi sắp xếp: 1356.

Câu 7: Cho công thức tính số tổ hợp m vật lấy từ n vật như sau:

$C(n,m) = 1$ nếu $m = 0$ hay $m=n$

$C(n, m) = C(n-1, m) + C(n-1, m-1)$ nếu $0 < m < n$

- Hãy viết hàm đệ quy C++ để tính $C(n,m)$.
- Viết chương trình C++ nhập vào hai số nguyên n và m và gọi hàm C định nghĩa ở câu a để tính $C(n,m)$ và trình bày kết quả ra.

Câu 8: Hãy viết hàm đệ quy để tìm chữ số đầu tiên của số nguyên dương n .

Bài tập thêm

Câu 1: Cho định nghĩa mảng và con trỏ sau:

```
int ara[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
int *ip1, *ip2;
```

Phép gán nào hợp lệ trong các phép gán sau:

- $ip1 = ara;$
- $ip2 = ip1 = \&ara[3];$
- $ara = 15;$
- $*(ip2 + 2) = 15;$ // Assuming $ip2$ and ara are equal.

Câu 2: Hãy chạy đoạn chương trình sau:

```
#include <iostream.h>  
void main()  
{  
    int num;  
    cout<< "Please enter the numbers of input: "  
    cin>>num;  
    int a = new int [num];  
    int total = 0; // Holds total of user's eight numbers.  
    int ctr;  
    for (ctr=0; ctr<num; ctr++)  
    {  
        cout << "Please enter the next number...";  
        cin >> a[ctr];  
        total += a[ctr];  
    }  
    cout << "The total of the numbers is " << total << "\n";  
    return;  
}
```

Câu 3: Chạy đoạn chương trình sau và giải thích

```

#include <iostream.h>
void swap(int a[], int *c1, int *c2, int *d1, int *d2);
void main()
{
    int a[2], c1, c2,d1,d2;
    int *x1, *x2, *y1, *y2;
    a[0] = 1 ; a[1] =2;
    c1 = 1; c2 =2;
    d1 = 1; d2 =2;

    x1 = &c1; x2 = &c2;
    y1 = &d1; y2 = &d2;

    swap(a, x1,x2,y1,y2);
    cout<<a[0]<<a[1]<<" "
        << *x1<<*x2<<" "
        <<*y1<<*y2;

    swap(a, x1,x2,y1,y2);
    cout<<a[0]<<a[1]<<" "
        << *x1<<*x2<<" "
        <<*y1<<*y2;
}
void swap(int a[], int *c1, int *c2, int *d1, int *d2)
{
    a[0] = 2 ; a[1] =1;
    *c1=2, *c2 =1;
    int* temp = d1;
    d1 =d2;
    d2 = temp;
}

```

Câu 4: Viết một hàm trả về căn bậc 4 của một đối số nguyên k . Giá trị trả về phải là một số chính xác kép. (Hướng dẫn: nên dùng hàm thư viện `sqrt()`).

Viết một chương trình C++ đọc vào một số nguyên n và gọi hàm nói trên tính căn bậc bốn của n .

Câu 5: Hãy viết hàm đệ quy để tìm ước số chung lớn nhất của hai số nguyên dương a và b

Câu 6: Cài đặt và minh hoạ bài toán tháp Hà Nội được trình bày bên dưới.

Bài toán **tháp Hà Nội** (tiếng Anh gọi là **Tower of Hanoi** hay **Towers of Hanoi**) xuất phát từ trò chơi đồ Tháp Hà Nội. Mục đích của bài toán là thực hiện được yêu cầu của trò chơi. Dạng bài toán thông dụng nhất là:

Người chơi được cho ba cái cọc và một số đĩa có kích thước khác nhau có thể cho vào các cọc này. Ban đầu sắp xếp các đĩa theo trật tự kích thước vào một cọc sao cho đĩa nhỏ nhất nằm trên cùng, tức là tạo ra một dạng hình nón. Người chơi phải di chuyển toàn bộ số đĩa sang một cọc khác, tuân theo các quy tắc sau:

- một lần chỉ được di chuyển một đĩa
- một đĩa chỉ có thể được đặt lên một đĩa lớn hơn (không nhất thiết hai đĩa này phải có kích thước liền kề, tức là đĩa nhỏ nhất có thể nằm trên đĩa lớn nhất)".



Bài toán này có lời giải chính xác. Tuy nhiên các mở rộng cho trường hợp có nhiều hơn ba cọc cho đến nay vẫn chưa được giải cận kề.

Thuật giải đệ quy

- đặt tên các cọc là A, B, C -- những tên này có thể chuyển ở các bước khác nhau
- gọi n là tổng số đĩa
- đánh số đĩa từ 1 (nhỏ nhất, trên cùng) đến n (lớn nhất, dưới cùng)

Để chuyển n đĩa từ cọc A sang cọc B thì cần:

1. chuyển $n-1$ đĩa từ A sang C. Chỉ còn lại đĩa # n trên cọc A
2. chuyển đĩa # n từ A sang B
3. chuyển $n-1$ đĩa từ C sang B cho chúng nằm trên đĩa # n

Phương pháp trên được gọi là thuật giải đệ quy: để tiến hành bước 1 và 3, áp dụng lại thuật giải cho $n-1$. Toàn bộ quá trình là một số hữu hạn các bước, vì đến một lúc nào đó thuật giải sẽ áp dụng cho $n = 1$. Bước này chỉ đơn giản là chuyển một đĩa duy nhất từ cọc A sang cọc B.