

Bài thực hành 1

1 Mục tiêu

- Hướng dẫn thống nhất phong cách lập trình C++ (sử dụng tab, ngoặc)
- Hướng dẫn cách nộp bài thực hành: sử dụng tài khoản bitbucket, quy định tên tài khoản (trùng tên tài khoản vnu), tên tệp (như trong đề bài thực hành) và repository (dsa01 cho bài thực hành 1, dsa02 cho bài thực hành 2 ...)
- Hướng dẫn nguồn tra cứu C++: <http://cplusplus.com/>, <http://www.parashift.com/c++-faq-lite/>
- Ôn tập ngôn ngữ C++

2 Bài tập

Câu 1 [gennum.cpp, readnum.cpp]

Sử dụng các luồng đọc/ghi tệp để làm những việc sau đây:

1. Viết một chương trình có tên gennum.cpp để sinh ra một chuỗi n số nguyên với n cũng là một số nguyên được sinh ngẫu nhiên. Viết chuỗi được sinh ra vào file tên DAYSO1.txt dưới cấu trúc như sau:

- Dòng đầu tiên chứa số n (là số các số nguyên)

- n dòng tiếp theo chứa n số nguyên, mỗi số nguyên được viết trên một dòng

2. Viết một chương trình có tên readnum.cpp để đọc một dãy số từ file DAYSO1.TXT dưới dạng sau:

- Dòng đầu tiên chứa số n là độ dài dãy số

- n dòng tiếp theo chứa n số nguyên, mỗi số nguyên được viết trên một dòng.

Ngoài ra, chương trình còn thực hiện những việc sau đây:

a) In ra số nguyên lớn nhất của dãy

b) In ra số nguyên lớn thứ hai của dãy

c) In ra số nguyên lớn thứ k của dãy, với k được đọc vào từ bàn phím

Câu 2 [reverse.cpp, cutk.cpp]

Cho file DAYSO2.TXT chứa các số nguyên, mỗi số được viết trên một dòng (số các số nguyên nhỏ hơn 10,000)

1. Viết một chương trình có tên reverse.cpp đọc tất cả các số nguyên từ file DAYSO2.TXT vào một mảng và ghi tất cả các số nguyên đó ra file KETQUA1.TXT theo thứ tự ngược lại với mỗi số nguyên được viết trên một dòng.

2. Giả sử n là số các số nguyên được ghi ở file DAYSO2.TXT. Viết một chương trình có tên cutk.cpp đọc vào tất cả các số nguyên từ file DAYSO2.TXT trong chuỗi và lấy ra k số ($0 < k \leq n$) (k được nhập vào từ bàn phím) ghi vào file KETQUA2.TXT sao cho mỗi số được viết trên một dòng và:

- (n-k) dòng đầu tiên ghi các số thứ k+1 đến n

- k dòng tiếp theo ghi các số thứ 1 đến k

Câu 3 [complex.cpp]

Viết một chương trình có tên complex.cpp định nghĩa lớp Complex và nạp chồng các toán tử: cộng, trừ, nhân. Lớp Complex thể hiện số phức. Một số phức có dạng $a + ib$, trong đó a và b là số thực, i là đơn vị ảo đặc trưng bởi $i^2 = -1$. Sau đây là các phép toán chuẩn của số phức:

$$(a + ib) + (c + id) = (a + c) + i(b + d)$$

$$(a + ib) - (c + id) = (a - c) + i(b - d)$$

$$(a + ib) * (c + id) = (ac - bd) + i(bc + ad)$$

3 Bài mẫu

Câu 1. Định nghĩa hàm `inOrder` nhận 3 đối số kiểu `int`. Hàm này trả về `true` nếu 3 đối số có giá trị giảm dần; ngược lại trả về `false`. Ví dụ `inOrder(1, 2, 3)` và `inOrder(1, 2, 2)` trả về `false`, `inOrder(4, 3, 2)` trả về `true`.

Trả lời

```
bool inOrder(int x, int y, int z){
    return (x >= y && y >= z);
}
```

Câu 2. Giải thuật Fisher-Yates là giải thuật sinh 1 hoán vị ngẫu nhiên của một tập hữu hạn phần tử. (Có thể ứng dụng vào bài toán trộn bài.) Giả mã của nó như sau:

Algorithm `shuffle(a, n)`
Input Một mảng `a` gồm `n` phần tử (đánh chỉ số từ 0 đến `n-1`)
Output Mảng `a` chứa hoán vị ngẫu nhiên của các phần tử trong `a` cũ
for `i ← n-1 to 1 do`
 `j ← số nguyên ngẫu nhiên với 0 ≤ j ≤ i`
 đảo `a[j]` và `a[i]`

Hãy viết lại hàm `shuffle` cài giải thuật Fisher-Yates này bằng ngôn ngữ C++ cho mảng các số nguyên.

Trả lời

```
void shuffle(int a[], int n){
    srand(time(0));
    for(int i = n - 1; i > 0; i--){
        int j = rand() % (i + 1);
        swap(a[i], a[j]);
    }
}
```

Câu 3. Định nghĩa hàm `main` đọc từ bàn phím 2 giá trị `m, n` nguyên dương; sau đó tiếp tục đọc `m x n` số nguyên dương có 2 chữ số vào mảng 2 chiều cấp phát động kích thước `m x n`. Nếu số nhập vào không phải số nguyên dương có 2 chữ số thì kết thúc chương trình. Chú ý giải phóng bộ nhớ động ở cuối hàm `main`.

Trả lời

```
int main()
{
    unsigned int m, n;
    // Nhập m, n
    cout << "Nhập m: ";
    cin >> m;
    cout << "Nhập n: ";
    cin >> n;
    // Cấp phát động
    int ** a;
    a = new int *[m];
    for(int i = 0; i < m; i++) a[i] = new int[n];
    // Nhập m x n số nguyên
    cout << "Nhập vào " << m * n << " số có 2 chữ số: " << endl;
    for(int i = 0; i < m; i++){
        for(int j = 0; j < n; j++){
            int x;
            cin >> x;
            if(x < 10 || x > 99) exit(EXIT_FAILURE);
            a[i][j] = x;
        }
    }
    // Giải phóng bộ nhớ động
    for(int i = 0; i < m; i++) delete [] a[i];
}
```

```
    delete [] a;
    a = NULL;

    return 0;
}
```

Câu 4. Tập INPUT.TXT là tập văn bản chứa các số nguyên cách nhau bởi 1 hoặc nhiều dấu trắng. Hãy viết chương trình đọc các số từ tập này và in ra màn hình tổng của chúng.

Trả lời

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

int main()
{
    ifstream fin("INPUT.TXT");
    int x;
    unsigned int sum = 0;
    while(fin >> x) sum += x;
    fin.close();
    cout << "Tong = " << sum << endl;
    cin.get();
    return 0;
}
```

Câu 5. Lớp HìnhVuong (biểu diễn hình vuông) thừa kế public từ lớp HìnhChuNhat (biểu diễn hình chữ nhật). Hãy viết định nghĩa hàm kiến tạo của HìnhVuong. Biết rằng hàm kiến tạo của HìnhChuNhat khởi tạo chiều dài và chiều rộng bằng giá trị các đối số.

```
class HìnhChuNhat
public:
    HìnhChuNhat(double d, double r);
    double dientich()const;
    double chuvi()const;
private:
    double chieudai;
    double chieurong;
```

```
class HìnhVuong: public HìnhChuNhat
public:
    HìnhVuong(double c);
```

Trả lời

```
HìnhVuong::HìnhVuong(double c): HìnhChuNhat(c, c){}
```