

Chủ đề: Thao tác với file trong C++

I. Thao tác với file là gì ?

- Thao tác với file là việc ghi, lấy và xử lý dữ liệu trong file khác để phục vụ cho việc chạy chương trình chính (C++)
- Trong chương trình học môn “Kỹ thuật lập trình cơ bản”, ta sẽ nghiên cứu về việc thao tác với file có đuôi .txt
- Để thực hiện các thao tác với tệp, trong C++ chúng ta sử dụng thư viện `<fstream>`

II. Các thao tác với file

1. Mở tệp tin

a) Mở tệp để ghi dữ liệu

- Khai báo dòng xuất tệp :
`ofstream <tên biến tệp>;` // biến f có vai trò là biến tệp\
VD: `ofstream f;`
- Mở tệp để ghi dữ liệu vào tệp:
`<tên_biến_tệp>.open(<tên_tệp.txt>,<Kiểu mở>;`
VD: `f.open("baitap.txt",ios::out)`
- Các kiểu mở phổ biến:
 - + `ios::out`: cho phép ta ghi đè dữ liệu
 - + `ios::app`: cho phép ghi bổ sung dữ liệu vào tệp (ghi chèn)
- Lưu ý:
 - + Nếu chưa tồn tại tệp txt trước đó, chương trình sẽ tự tạo file và thao tác trên file đã tạo. Ngược lại sẽ thao tác trên file đã có sẵn theo kiểu mở đã code
 - + Tên tệp có thể là đường dẫn dẫn đến tệp (đường dẫn tuyệt đối). Nếu viết theo đường dẫn, file sẽ được lưu trong máy tính theo đường dẫn đó

VD: D:/baitap.txt -> file sẽ được lưu trong ổ D

Tuy nhiên nếu chỉ viết tên file (VD: baitap.txt – đường dẫn tương đối) -> file sẽ được lưu tại tệp đang chứa chương trình C++

Mặc định trong thời gian tới, chúng ta sẽ sử dụng đường dẫn tương đối

VD: Mở và ghi đè nội dung vào file có tên baitap.txt

```
ofstream f;
```

```
f.open("baitap.txt",ios::out);
```

- Đặc biệt, chúng ta có thể viết gộp 2 câu lệnh trên thành:

```
ofstream f("baitap.txt",ios::out);
```

b) Mở tệp để đọc dữ liệu:

- Tệp phải tồn tại trước đó mới có thể đọc được.
- Khai báo dòng nhập vào tệp:

```
ifstream <Tên biến tệp>;
```

- Mở tệp để đọc:

```
<Tên biến tệp>.open(<tên tệp>,ios::in);
```

- Chúng ta cũng có thể viết gộp 2 dòng lệnh này như sau:

```
ifstream <tên biến tệp> (<tên tệp>,ios::in)
```

c) File vừa có thể ghi dữ liệu, vừa có thể đọc dữ liệu:

- Ta sử dụng cú pháp

```
fstream <tên biến tệp> (<tên tệp>, <kiểu mở/đóng>);
```

- Trong trường hợp bạn muốn vừa mở file để đọc và mở file để ghi, bạn sẽ khai báo như sau:

```
fstream <tên biến tệp> (<tên tệp>, ios::in | ios::out); // ghi chèn  
hoặc fstream <tên biến tệp> (<tên tệp>, ios::in | ios::app); // ghi đè
```

d) Đóng tệp:

- Sau các quá trình làm việc với file dữ liệu, ta phải đóng file dữ liệu:

Cú pháp:

```
<tên biến tệp>.close();
```

e) Kiểm tra file có mở được hay không:

- Trong các bài tập, bạn cần kiểm tra xem file có mở được hay không:
- Cú pháp <tên biến tệp>.is_open();
- Hàm sẽ trả về true nếu mở được, và false nếu không.

2. Ghi dữ liệu vào tệp

- Ta sử dụng cú pháp như sau để ghi dữ liệu vào tệp
<tên biến tệp> << <nội dung ghi>

VD1: Ghi dòng chữ “Hello !” vào tệp có tên là baitap.txt, ta làm như sau:

```
#include <iostream>
#include <fstream>

using namespace std;

int main(){
    fstream f("baitap.txt",ios::out);
    if (!f.is_open()){
        cout << "Tệp không mở được !";
        return 0;
    }
    f << "Hello !" << endl;
    f.close();
}
```

VD2: Ghi ra file “so.txt” các số trong khoảng từ 1 đến 10, mỗi số cách nhau bởi 1 dấu cách:

```
#include <iostream>
#include <fstream>

using namespace std;
```

```

int main(){
    fstream f("so.txt",ios::out);

    if (!f.is_open()){
        cout << "Tập không mở được !";
        return 0;
    }

    for (int i = 1; i <= 10; i++)
        f << i << " ";
    f.close();
}

```

Bài tập: (Trong các bài tập dưới đây, mảng có thể cấp phát tĩnh hoặc động. Có thể sử dụng cấu trúc hàm)

Bài 1: Nhập 1 mảng a có n phần tử, n và giá trị các phần tử được nhập vào từ bàn phím. Tính tổng tất cả các phần tử âm trong mảng đó. Xuất kết quả ra file có tên "bai1.txt"

Bài 2: Nhập 3 số nguyên dương a,b,c. Xuất ra file có tên "bai2.txt" trung bình cộng của 3 số đó

Bài 3: Nhập 1 mảng a có n phần tử, n và giá trị các phần tử được nhập vào từ bàn phím. Sắp xếp mảng đó theo thứ tự giảm dần. Xuất kết quả ra file có tên "bai1.txt"

Bài 4: Nhập vào 1 chuỗi ký tự a. Xuất ra file "bai3.txt" các kết quả sau:

- Độ dài của chuỗi a
- Số lần xuất hiện ký tự là "b" trong chuỗi

- Số từ trong a
- Chuỗi a nhưng đã sắp xếp các cụm từ trong chuỗi theo thứ tự tăng dần trong từ điển

Bài 5: Nhập 1 mảng a có n phần tử (n và các phần tử được nhập vào từ bàn phím).
Xuất ra file “bai5.txt” tần suất xuất hiện của các phần tử trong mảng:

VD:

Phần tử 2 xuất hiện 3 lần

Phần tử 3 xuất hiện 4 lần

3. Đọc dữ liệu từ tệp

Có 2 cách đọc dữ liệu từ tệp (tệp phải có sẵn)

a) Đọc từng dòng trong tệp

- Khi đọc từng dòng, mỗi dòng được xem là **1 chuỗi ký tự**. Do vậy cách này sẽ không phù hợp để nhập dữ liệu số, và ta phải tự tay tách hoặc sử dụng những biện pháp khác để đọc. Việc đọc từng dòng trong tệp sẽ phù hợp với nhu cầu đọc tệp thuần văn bản
- Cú pháp:

`getline(<tên file>,<tên biến nhận dữ liệu>);`

- Trong quá trình ghi file, để kiểm tra xem đã đọc hết tệp hay chưa, ta sử dụng phương thức eof(). Nó sẽ trả về true nếu như file đã được đọc hết (con trỏ đã đọc tới cuối trang), và false nếu chưa đọc hết. Điều này sẽ hữu dụng khi bạn đọc dữ liệu từ file có nhiều dòng và nhiều dữ liệu

VD: Đọc từ file “ten.cpp” tên của 4 bạn học sinh (mỗi tên viết trên 1 dòng) và lưu vào mảng a. In các phần tử của mảng a ra màn hình, mỗi phần tử nằm trên 1 dòng:

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```
int main(){  
    fstream f("ten.txt",ios::in);  
    if (!f.is_open()){  
        cout << "Tập không mở được !";  
        return 0;  
    }  
    string s[4];  
    int i = 0;  
    while (!f.eof()){  
        getline(f, s[i]);  
        i++;  
    }  
    for (int i = 0; i < 4; i++)  
        cout << s[i] << endl;  
    f.close();  
}
```

b) Đọc từng cụm trong tệp:

- Thường áp dụng trong những dữ liệu kiểu số, các cụm cách nhau bởi dấu cách hoặc xuống dòng.
- Cú pháp <tên biến tệp> >> <tên biến nhận dữ liệu>;

VD: Cho tệp “mang.txt” có các dữ liệu sau:

- Dòng đầu tiên chứa 1 số tự nhiên nguyên dương, chỉ số lượng phần tử trong mảng
- Dòng thứ 2 chứa các phần tử là số tự nhiên trong mảng

Hãy đọc dữ liệu của tệp và lưu vào 1 biến mảng a, và in mảng ra màn hình cùng với tổng của các phần tử trong mảng

Bài làm:

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```
int main(){
```

```
    fstream f("ten.txt",ios::in);
```

```
    if (!f.is_open()){
```

```
        cout << "Tep khong mo duoc !";
```

```
        return 0;
```

```
    }
```

```
    int n;
```

```
    f >> n;
```

```
    int a[n], tong = 0;
```

```
    for (int i = 0; i < n; i++){
```

```
        f >> a[i];
```

```

        tong += a[i];
    }
    for (int i = 0; i < n ; i++)
        cout << a[i] << " ";
    cout << endl << "Tong cua cac phan tu la: " << tong << endl;
    f.close();
}

```

Bài tập tổng hợp:

Sử dụng cấu trúc hàm, mảng, con trỏ và thao tác với file, cũng như những cấu trúc lập trình khác, hãy làm các bài tập sau. Mỗi gạch đầu dòng sử dụng tối thiểu 1 hàm, hàm main chạy toàn bộ các hàm của chương trình trên

Câu 1:

- Nhập mảng a gồm n phần tử số nguyên (n và các phần tử nhập từ bàn phím)
- Xuất dãy số vừa nhập ra màn hình và ghi vào tệp với tên tệp có tên "bai1.txt"
- Sắp xếp mảng a theo thứ tự giảm dần và xuất chèn vào tệp "bai1.txt"
- Gọi P là tổng các phần tử dương trong mảng a và q là trung bình cộng của các phần tử trong dãy. Tính $k = p/q$, in kết quả ra màn hình
- Cho biết tổng của số lớn nhất và số nhỏ nhất trong mảng a, in kết quả ra màn hình và ghi chèn vào file "bai1.txt"

Câu 2:

- Nhập mảng a gồm n phần tử số nguyên dương (n và các phần tử nhập từ bàn phím, $n > 15$)
- Hiển thị ra màn hình các phần tử chẵn trong mảng
- Đếm tần suất xuất hiện của các phần tử trong mảng. In toàn bộ kết quả ra file "bai2.txt" theo mẫu câu sau
"Phan tu <giá trị của phần tử> xuất hiện <số lần xuất hiện> lần."
- Nhập vào 1 số nguyên x, đếm xem có bao nhiêu phần tử là ước của x. Ghi chèn vào file "bai2.txt"
- Kiểm tra phần tử lớn nhất trong mảng có phải số chính phương hay không ? In ra màn hình kết quả "YES" hoặc "NO"

Câu 3:

- Nhập mảng A gồm n số thực (n và mảng nhập từ bàn phím)
- Xuất dãy số vừa nhập và ghi vào tệp có tên "bai3.txt"
- Sắp xếp dãy số A theo thứ tự tăng dần của các số nguyên tố trong mảng(giữ nguyên vị trí của các phần tử không phải là số nguyên tố). Bổ sung vào file "bai3.txt"
- Tìm xem có bao nhiêu phần tử nhỏ nhất trong mảng
- Nhập số x từ bàn phím, xóa hết tất cả các phần tử nhỏ hơn x ra khỏi mảng. Ghi bổ sung vào file "bai3.txt"

Câu 4:

- Nhập mảng A gồm n số thực (A và n được đọc từ file "bai4.cpp", khi làm sẽ được gửi file)
- Tính trung bình cộng của các phần tử trong mảng. Ghi đè vào file "bai4.cpp"
- Xóa hết các phần tử là số nguyên tố ra khỏi mảng. In mảng ra màn hình và ghi chèn vào file "bai4.cpp"

(Phần bài tập còn lại là các bài tập đã làm trong suốt quá trình học từ đầu đến giờ.
Tập cách ghi và đọc các phần tử trong quá trình chạy chương trình)