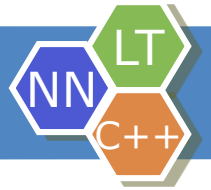
TRƯỜNG ĐẠI HỌC KINH TẾ KỸ THUẬT CÔNG NGHIỆP
Khoa Công Nghệ Thông Tin

FUTURE TECHNOLOGY
CÔNG NGHỆ TƯƠNG LAI

Phần 2. NN lập trình C++

CHƯƠNG 10
TỆP (FILES)

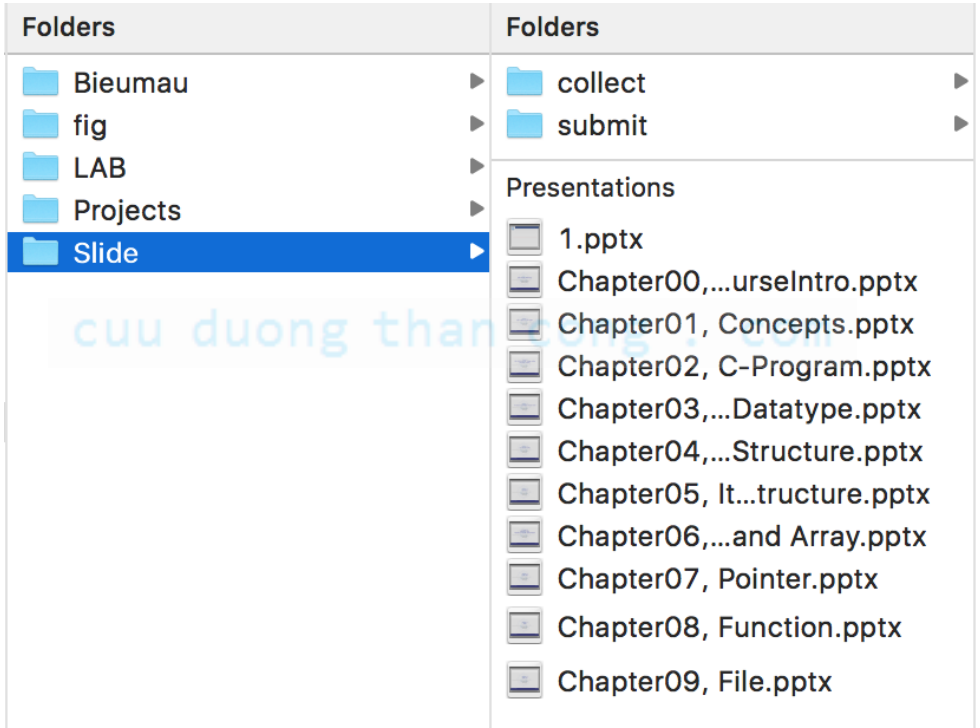


Chương 10 – Tệp (Files)

1. Giới thiệu tệp
2. Tệp văn bản
3. Tệp nhị phân

10.1 Giới thiệu tệp

- Khi một chương trình kết thúc thực thi, các biến dữ liệu liên quan sẽ bị dọn dẹp khỏi bộ nhớ chính (RAM) của máy tính
  - => Để dữ liệu không bị chương trình mất đi khi chương trình kết thúc, chương trình cần lưu chúng dưới dạng tập tin (file) vào các thiết bị lưu trữ như ổ cứng, CD, DVD, v.v.

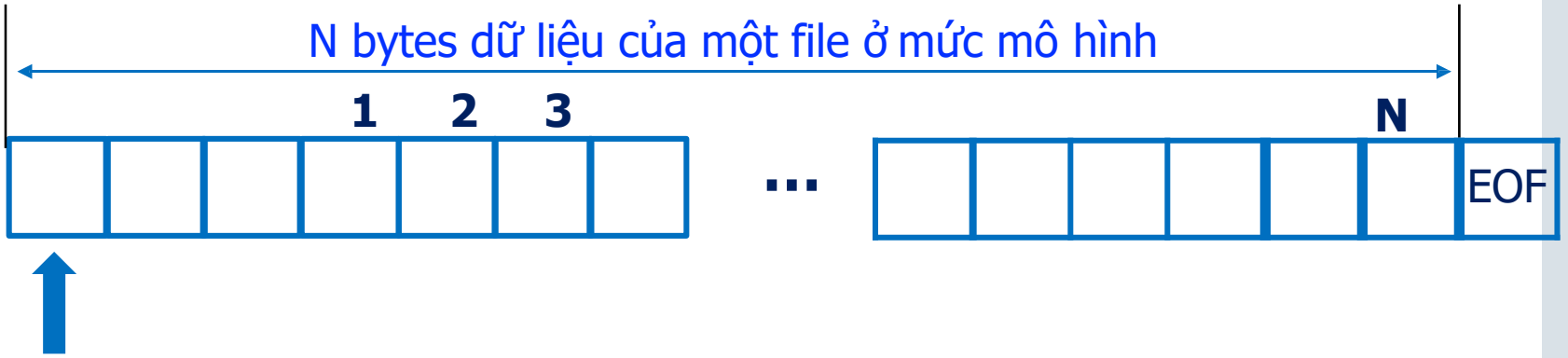


10.1 Giới thiệu tệp

- Tập tin là một dãy các bytes dữ liệu, như hình vẽ, kết thúc bằng ký hiệu đặc biệt EOF
  - EOF (End Of File): là giá trị đặt biệt, không trùng với bất cứ giá trị của byte dữ liệu nào.
  - EOF: Ký hiệu mà các hàm đọc dữ liệu trả về để cho biết kết thúc tập tin.
  - (Nhiều hệ thống EOF = -1)

10.1 Giới thiệu tệp

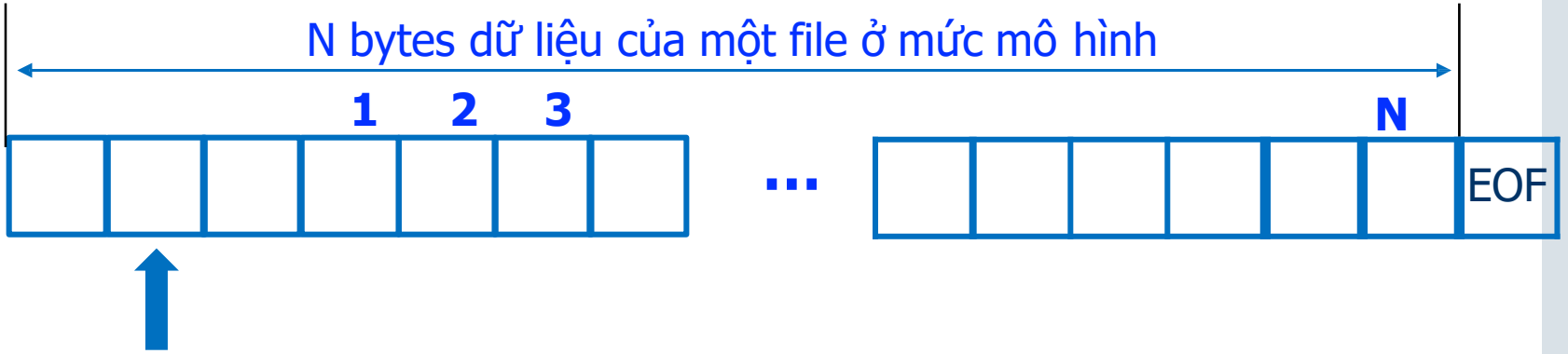
Thẻ đánh dấu trong tập tin



Sau khi mở tập tin thành công, thẻ đánh dấu tự động chỉ đến byte đầu tiên của tập tin

10.1 Giới thiệu tệp

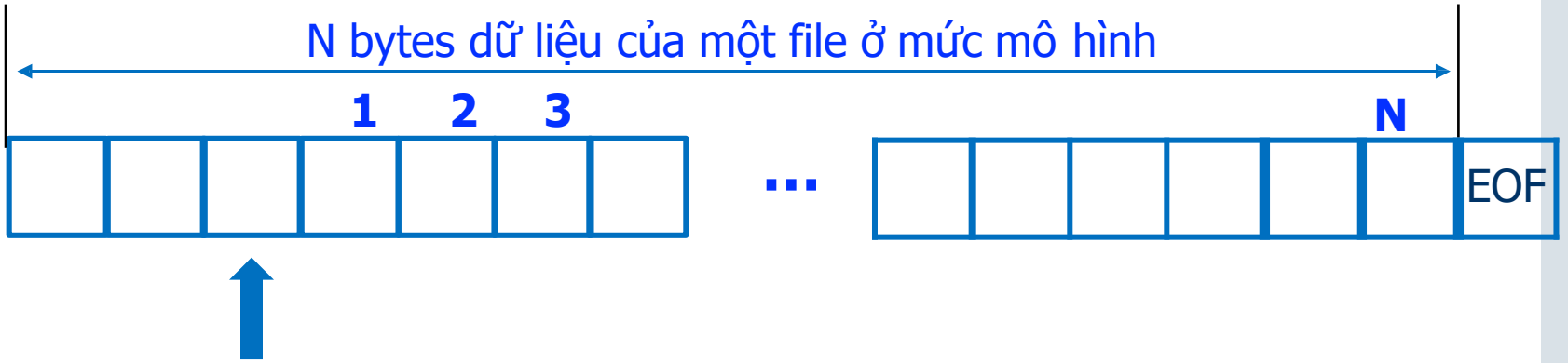
Thẻ đánh dấu trong tập tin



Sau khi đọc 1 byte dữ liệu, ví dụ sử dụng hàm fgetc

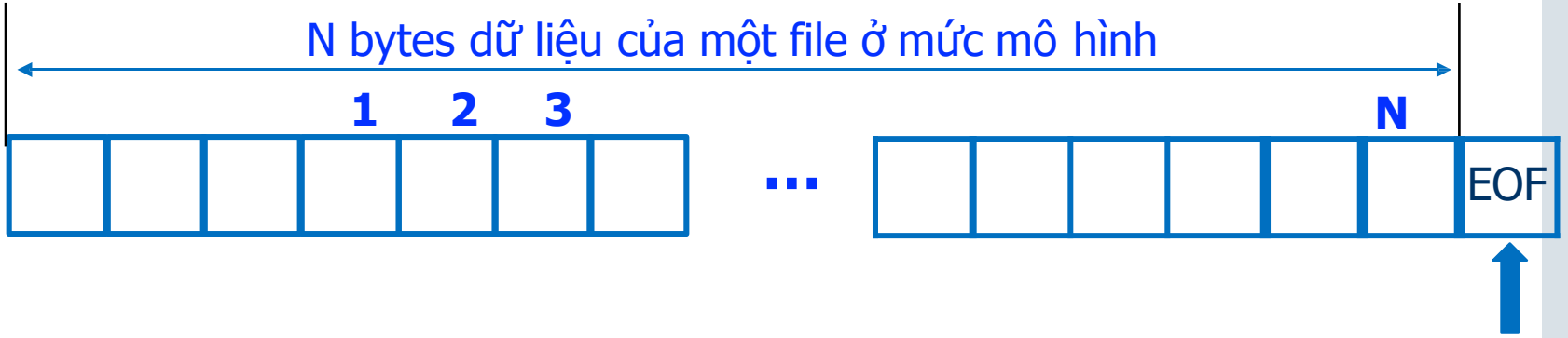
10.1 Giới thiệu tệp

Thẻ đánh dấu trong tập tin



Sau khi đọc 1 byte dữ liệu nữa

Thẻ đánh dấu trong tập tin



Sau khi đã đọc thành công N bytes
Thẻ đánh dấu chỉ đến EOF

Lần đọc dữ liệu kế tiếp hàm đọc trả về giá trị EOF để nói rằng kết thúc tập tin, và giá trị trả về là EOF (-1)

10.1 Giới thiệu tệp

- **Phân loại: 2 loại**
  - Tệp văn bản (text)
  - Tệp nhị phân (binary)
- **Tệp văn bản:**
  - Lưu trữ: các ký tự nhị phân được mã hóa dạng mã ASCII: 8bit = 1 byte = 1 ký tự.
  - Mở: có
  - Đọc: Đọc được khi mở bằng trình đọc file.
  - Người dùng dễ kiểm tra dữ liệu trong tệp tuy nhiên khó xử lý mới máy tính.
- **Tệp nhị phân:**
  - Lưu trữ: các ký tự mã hóa dưới dạng số nhị phân.
  - Mở: có
  - Đọc : Không đọc được vì dữ liệu là các ký hiệu đặc biệt
  - Người dùng không kiểm tra được dữ liệu tuy nhiên dễ xử lý với máy tính.

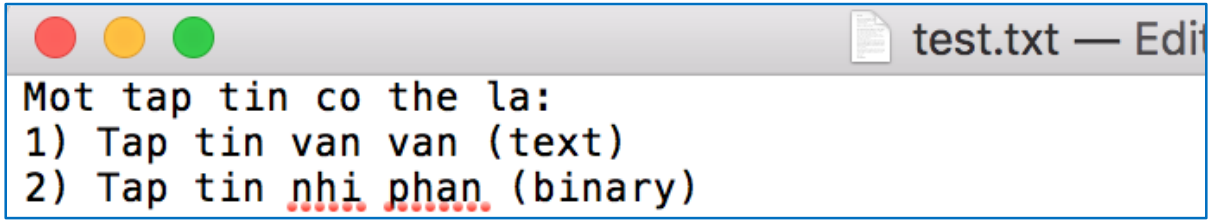
10.1 Giới thiệu tệp

Trong C++, khi thao tác với một tệp dữ liệu, cần thực hiện tuần tự theo các bước như sau:

- (1) Khai báo sử dụng kiểu dữ liệu tập tin
- (2) Mở tập tin
  - Hàm: **fopen, nói sau**
- (3) Thao tác với tập tin
  - Đọc hay ghi dữ liệu
  - Mỗi lần đọc hay ghi dữ liệu, thẻ đánh dấu trong tập tin tự động tăng đến phần tử tiếp theo
- (4) Đóng tập tin
  - Hàm: **fclose, nói sau**

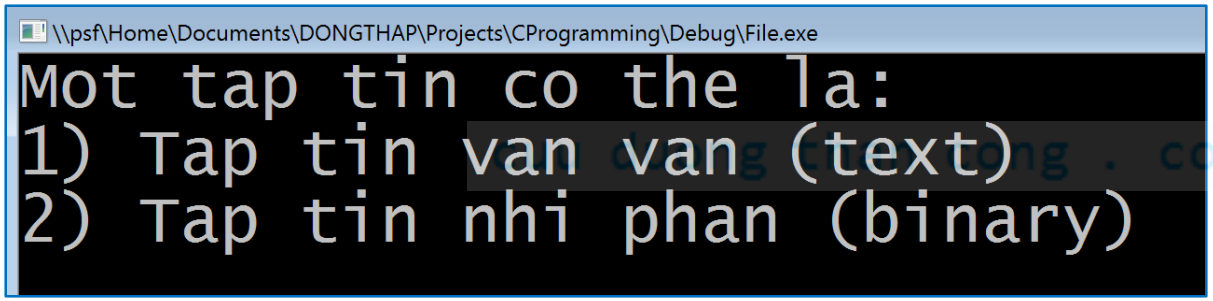
Thư viện cần mở: **#include<fstream.h>**

10.2 Tệp văn bản



Vào: tập tin văn bản (đọc được)

Chương trình: đọc các dòng văn bản và đưa vào bộ đệm

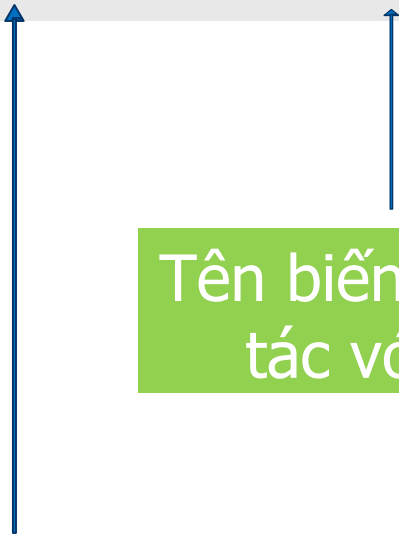


Ra: in lại các dòng trên màn hình

10.2 Tệp văn bản

(1) Khai báo biến tệp

```
<fstream>    filevar;
```



Tên biến tệp, để thực hiện các thao tác với tệp thông qua tên này

Kiểu dữ liệu **fstream**
Định nghĩa trong <fstream.h>
➔ Đặt #include <fstream.h> đầu chương trình

10.2 Tệp văn bản

(2) Mở tệp

Dùng hàm thiết lập:

<fstream> filevar(<filename>, <i/o mode>);

Hoặc dùng hàm thành phần open của đối tượng luồng nhập/xuất:

<fstream> filevar;

filevar.open(<filename>, <i/o mode>);

Trong đó:

- *<fstream>:* fstream – Mở tệp chung ;
- *filevar* - biến tệp: dùng để thực hiện các thao tác với tệp gắn với nó.
- *filename* : là tên tệp dữ liệu cần thao tác trên nó.
- *i/o mode* : là chế độ mở tệp tin, là các hằng kiểu bít đã được định nghĩa sẵn bởi C++.

10.2 Tệp văn bản

(2) Mở tệp

Mở tệp để ghi:

ofstream filevar(<filename>);

Ví dụ: ***ofstream f("abc.txt");*** ***// mở tệp để ghi***

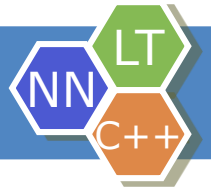
Mở tệp để đọc:

ifstream filevar(<filename>);

Ví dụ: ***ifstream f("dulieu.txt");*** **//mở tệp để đọc**

Trong đó:

- *ofstream* – mở chỉ để ghi; *ifstream* – mở chỉ để đọc
- *filevar* - biến tệp: dùng để thực hiện các thao tác với tệp gắn với nó.
- *filename* : là tên tệp dữ liệu cần thao tác trên nó.



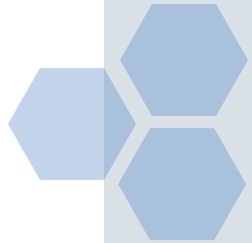
10.2 Tệp văn bản

(2) Mở tệp

❖ Các chế độ mở tệp tin

Các chế độ mở tệp tin được định nghĩa bởi các bit chỉ thị:

- **ios::in**: Mở một file để đọc.
- **ios::out**: Mở một file có sẵn để ghi.
- **ios::app** Mở một file có sẵn để thêm dữ liệu vào cuối tệp.
- **ios::binary**: Mở một file ở chế độ nhị phân.



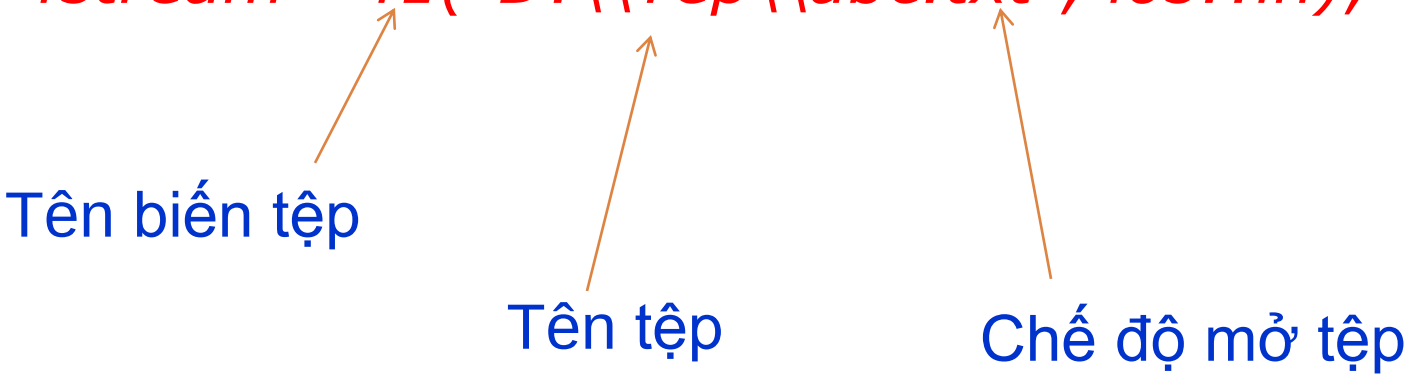
10.2 Tệp văn bản

- **Chú ý:**
  - Có thể kết hợp nhiều chế độ mở tệp:
    - Kết hợp bằng " |" ios::binary | ios::out
  - Chỉ mở tệp ở một trong hai chế độ: văn bản hoặc phị phân
  - Không nên mở tệp đồng thời với hai chế độ vừa ghi vừa đọc, nên tách thành 2 lần mở tệp riêng.

10.2 Tệp văn bản

- **Ví dụ:**

```
fstream    f ("sohoc.txt", ios::in);
fstream    f1("D:\\Tep\\abc.txt", ios::in);
```



Chú ý: Tệp nằm trong thư mục khi viết đường dẫn đến thư mục đó sử dụng "\\" thay cho "\"

10.2 Tệp văn bản

Ví dụ:

```
fstream   f;
f.open("dulieu.txt",ios::out);   //mở tệp văn bản để ghi
fstream   f("xyz.txt",ios::in);  //mở tệp văn bản để đọc

ofstream  f("abc.txt");     // mở tệp để ghi
ifstream  f("dulieu.txt");  // mở tệp để đọc
```

10.2 Tệp văn bản

(3) Thao tác với tệp tin

- ***Ghi tệp dữ liệu vào tệp văn bản bằng "<<"***

  filevar << Dữ liệu;

- ***Đọc dữ liệu từ tệp văn bản bằng ">>"***

  filevar >> Biến dữ liệu;

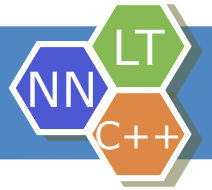
- ***Kiểm tra việc mở tệp***

Trước khi đọc/ghi file, cần kiểm tra file đã được mở thành công hay không:

- if (!filevar) … else …

Trong quá trình làm việc với file, cần dùng hàm kiểm tra kết thúc tập tin trong vòng lặp:

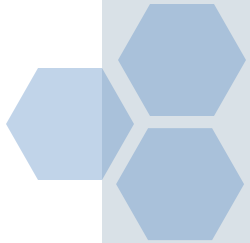
- while(!filevar.eof()) …



10.2 Tệp văn bản

(4) Đóng tệp

```
filevar.close();
```



10.2 Tệp văn bản

Ví dụ 1: Ghi vào tệp văn bản dãy các số chẵn từ 1->100

```
#include<fstream.h>
#include<iostream.h>
void GhiSoChan(ofstream &file)
{
    int dem = 0;
    file<<"Day so chan tu 1 -> 100 \n";
    for(int a = 1; a <= 100; a++)
    {
        if(a%2 == 0)
        {
            dem ++;
            file<<a;
        }
       if(dem % 10 == 0)
                file<<"\n";
       if(dem % 10 != 0)
                file<<"\t";
     }
    cout<<"Da ghi tep";
}
```

```
int main()
{
ofstream f("so_chan.txt");
if(!f)
{ cout<<"Khong the ghi tep";
  exit(1);
}
GhiSoChan(f);
f.close();

return 0;
}
```

10.2 Tệp văn bản

- **Ví dụ 2:** Cho trước tệp văn bản "in.txt"chứa 2 số nguyên. Viết các hàm để đọc thông tin từ tệp, tìm ước chung lớn nhất của hai số. Kết quả ghi vào tệp "out.txt"

10.2 Tệp văn bản

```
#include <iostream.h>
#include <fstream.h>
void doctep(char tentep[], int &n, int &m)
{   fstream  f(tentep,ios::in);
    if(!f)
    {   cout << "Khong the tao duoc tep tin "<<f<<endl;
        exit(1);
    }
    f>>n>>m;
    f.close();
}
int UCLN(int n, int m)
{
        while(m!=n)
                if(m>n) m=m-n;
                else
                        n=n-m;
     return n;
}
```

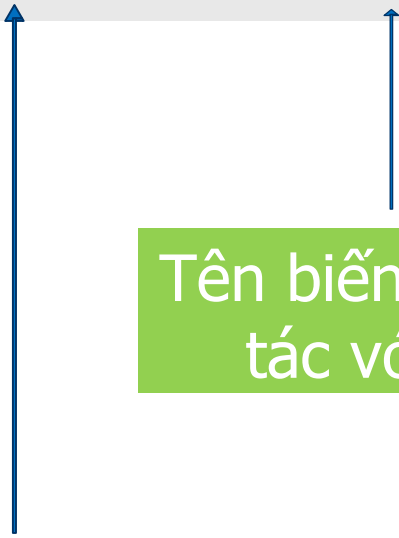
10.2 Tệp văn bản

```
void ghitep(char tentep[], int &n, int & m)
{
      fstream  f1(tentep,ios::out);
      if(!f1)
      { cout<<"Khong the tao duoc tep tin"<<f1<<endl;
        exit(1);
      }
      f1<<"\n"<<"UCLN = "<<UCLN(n,m);    //ghi vao tep
}
int main()
{ char         f[30];
  int          n, m;
  cout<<"\n Nhap ten tep doc du lieu: "; cin.getline(f,30);
  doctep(f, n, m);
  cout<<"\n Nhap ten tep ghi du lieu"; cin.getline(f,30);
  ghitep(f, n, m);
  return 0;
}
```

10.3 Tệp nhị phân

(1) Khai báo biến tệp

```
<fstream>    filevar;
```



Tên biến tệp, để thực hiện các thao tác với tệp thông qua tên này

Kiểu dữ liệu **fstream**
Định nghĩa trong <fstream.h>
➔ Đặt #include <fstream.h> đầu chương trình

10.3 Tệp nhị phân

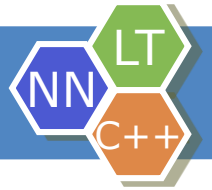
(2) Mở tệp

fstream *<Tên biến tệp>(<Tên tệp>, <Chế độ mở tệp>);*

ifstream *<Tên biến tệp>(<Tên tệp>, <Chế độ mở tệp>);*

ofstream *<Tên biến tệp>(<Tên tệp>, <Chế độ mở tệp>);*

- **Tên biến tệp**: dùng để thực hiện các thao tác với tệp gắn với nó.
- **Tên tệp**: là tên tệp dữ liệu cần thao tác trên nó.
- **Chế độ mở tệp**: là các hằng kiểu bít đã được định nghĩa sẵn bởi C++. Nó chỉ ra rằng ta đang mở tệp tin ở chế độ nào: đọc hoặc ghi, hoặc cả đọc lẫn ghi.

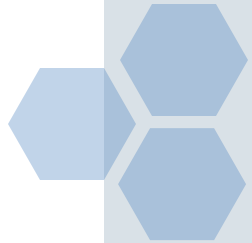


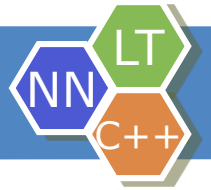
10.3 Tệp nhị phân

Ví dụ:

```
ofstream f("abc.dat", ios::binary);     //mở tệp nhị phân để ghi
ifstream f("dulieu.dat", ios::binary);  //mở tệp nhị phân để đọc
```

```
fstream   f;
f.open("dulieu.txt",ios::out | ios::binary);
fstream   f("xyz.txt",ios::in | ios::binary);
```





10.3 Tệp nhị phân

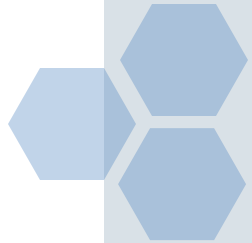
(3) Thao tác với tệp tin

- ***Ghi dữ liệu vào tệp nhị phân***

<Tên biến tệp>.write(char* <Dữ liệu ra>,int <Kích thước dữ liệu>);

- ***Đọc dữ liệu từ tệp nhị phân***

<Tên biến tệp>.read(char* <Dữ liệu>,int <Kích thước dữ liệu>);



10.3 Tệp nhị phân

- Trong đó, thao tác **write, read** nhận hai tham số đầu vào:

write(char* <Dữ liệu>,int <Kích thước dữ liệu>);

- Tham số thứ nhất: là con trỏ kiểu char trỏ đến vùng dữ liệu cần ghi vào tệp. Vì con trỏ bắt buộc có kiểu char nên khi muốn ghi dữ liệu có kiểu khác vào tệp sẽ dùng hàm ép kiểu:
  - *reinterpret_cast<char *>(<Dữ liệu>);*
  - *(char *)(dữ liệu);*
- Tham số thứ hai là kích cỡ dữ liệu được ghi vào tệp. Kích cỡ này được tính theo byte, nên dùng toán tử:

sizeof(<Kiểu dữ liệu>);

10.3 Tệp nhị phân

10.3.4 Kiểm tra mở tệp

Trước khi đọc/ghi file, cần kiểm tra file đã được mở thành công hay không:

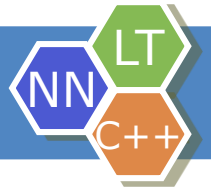
- if (!filevar) … else …

Trong quá trình làm việc với file, cần dùng hàm kiểm tra kết thúc tập tin trong vòng lặp:

- while(!filevar.eof()) …

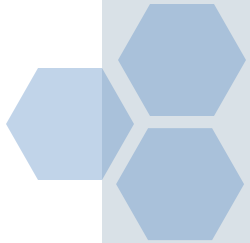
10.3.5 Đóng tệp

filevar.close();



10.3 Tệp nhị phân

- **Ví dụ:** Ghi dữ liệu một mảng vào tệp nhị phân, sau đó đọc dữ liệu mảng đã ghi từ tệp đã ghi và in ra màn hình?



10.3 Tệp nhị phân

```
#include <iostream.h>
#include <fstream.h>
const int SIZE = 10;
int  main()
{
   fstream    file;
   int a[SIZE] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10},  b[SIZE];
// Ghi du lieu
   file.open("a.dat", ios::out | ios::binary);
    if(!file){
       cout << "Khong the tao duoc tep tin " << file <<
endl;  exit(1);
    }
   file.write((char *)(a), sizeof(a));    // ghi du lieu
   vao tep
     file.close();
```

10.3 Tệp nhị phân

```
// Doc du lieu
file.open("a.dat", ios::in | ios::binary);
 if(!file)
{     cout << "Khong the tao duoc tep tin " << file
<< endl;    exit(1);
}
   file.read((char *)(b), sizeof(b));          // doc
   du lieu tu tep
for (int i = 0; i < SIZE; i++)
      cout << b[i] << " ";
cout << endl;
file.close();
return 0;
}     //end main
```

Câu hỏi lý thuyết

1. Nêu cách sử dụng tệp văn bản? Cho ví dụ.
2. Nêu cách sử dụng tệp nhị phân? Cho ví dụ.

Bài tập

Bài 1: Cho tệp văn bản "in1.txt"chứa ba số nguyên dương. Viết hàm đọc thông tin từ tệp, tìm số lớn nhất, nhỏ nhất. Kết quả ghi vaò tệp văn bản "out1.txt"?

Bài 2: Cho tệp văn bản **"in1.txt"**chứa 3 số nguyên dương. Viết hàm đọc thông tin từ tệp để sử dụng 3 số làm tham số giải phương trình bậc 2: $ax^2+bx+c=0$. Thông tin nghiệm ghi vaò tệp văn bản **"out1.txt"**?

Bài 3: Cho tệp văn bản "in1.txt"chứa 3 số nguyên dương. Viết hàm đọc thông tin từ tệp kiểm tra 3 số có là 3 cạnh tam giác không, tính chất tam giác. Kết quả ghi vào tệp văn bản "out1.txt"?

Bài tập

Bài 4: Cho tệp văn bản "in2.txt" có: dòng đầu tiên chứa số phần tử trong dãy, hàng thứ 2 chứa giá trị các phần tử trong dãy. Viết hàm đọc thông tin từ tệp thực hiện:

a) Tìm số nguyên tố, hoàn hảo, số đối xứng. Kết quả ghi vào cuối tệp văn bản "in2.txt"

b) Sắp xếp mảng tăng dần sử dụng Interchange sort? Kết quả ghi vào tệp văn bản "out2.txt"

Ví dụ: Ban đầu tệp "in2.txt" chứa:

5

28 17 70 31 11

Sau khi thực hiện chương trình tệp "in2.txt"

5

28 17 70 31 11

So nguyen to la : 17 11 31

So hoan hao la: 28

So doi xung la: 11

Bài tập

Bài 5: Cho tệp văn bản **"in2.txt"** có: dòng đầu tiên chứa số phần tử trong dãy, hàng thứ 2 chứa giá trị các phần tử trong dãy.

Viết hàm đọc thông tin từ tệp thực hiện:

a) Đếm phần tử chia hết cho 7, số nguyên tố đầu tiên, đếm số giá trị lớn nhất trong mảng. Kết quả ghi vào cuối tệp văn bản "**in2.txt**"

b) Sắp xếp mảng giảm dần sử dụng selection sort? Kết quả ghi vào tệp văn bản **"out2.txt"**

Ví dụ: Ban đầu tệp "in2.txt" chứa:

5

21 30 5 30 10

Sau khi thực hiện chương trình tệp "in2.txt"

5

21 35 5 35 10

So phan tu chia het cho 7 la : 3

So nguyen to dau tien la: 5

So gia trị lon nhat :2

Bài tập

Bài 6: Cho tệp văn bản **"hocsinh.txt"**. chứa thông tin: dòng 1 chứa số học sinh, từ dòng 2 chứa thông tin từng học sinh. Biết cấu trúc **HocSinh** gồm: **Mã học sinh** (char mahs[15]), **tên (**char tenhs[15]), **điểm trung bình** (float tb), **hạnh kiểm** (char hk[10]).

Viết chương trình thực hiện các chức năng sau**:**

a) Đọc thông tin từ tệp và hiển thị danh sách học sinh

b) Đếm số học sinh tên là "Trung" hạnh kiểm "TB" hoặc học sinh tên là "Anh" có điểm trung bình >8.0. Kết quả ghi vào tệp "kq.txt"