



**FUTURE TECHNOLOGY**  
CÔNG NGHỆ TƯƠNG LAI

# **ĐẠI HỌC KINH TẾ KỸ THUẬT CÔNG NGHIỆP**

## **KHOA CÔNG NGHỆ THÔNG TIN**

# **Bài Thực Tập 14**

## **Lập trình với biến tệp**

**Hà Nội - 2022**

# NỘI DUNG

1

**Mục tiêu bài học**

2

**Hướng dẫn học tập**

3

**Nội dung bài học**

4

**Giao nhiệm vụ tuần tiếp theo**

# MỤC TIÊU BÀI HỌC

- **Mục tiêu:**

Trang bị cho sinh viên kỹ năng lập trình cơ bản trong C++:

- + Khai báo tệp

- + Vào ra với tệp văn bản

**Kết quả đạt được:**

Sinh viên thành thạo các kỹ thuật lập trình với **tệp văn bản**, áp dụng giải các bài tập từ đơn giản đến phức tạp.

# HƯỚNG DẪN HỌC TẬP

**Để hoàn thành tốt bài học này sinh viên cần thực hiện những nhiệm vụ sau:**

- Đọc trước tài liệu: “B14\_Tailieu\_TTLTCB” phần nhắc lại kiến thức lý thuyết mục “E. TÓM TẮT LÝ THUYẾT”
- Cài đặt, sử dụng được công cụ thực hành Cfree 5.0.
- Thực hành trên máy tính các bài thực hành mẫu.
- Hoàn thành các bài thực hành tự làm cuối bài học.
- Hoàn thành các bài tập giao về nhà.
- Trao đổi, thảo luận với giảng viên qua các phương thức:
  - + Thảo luận đặt câu hỏi trên diễn đàn.

# NỘI DUNG BÀI HỌC

## I. Hướng dẫn ban đầu (90 phút)

- 1.1 Tổng hợp lý thuyết
- 1.2 Hướng dẫn thực hành bài số 01
- 1.3 Hướng dẫn thực hành bài số 02

## II. Hướng dẫn thường xuyên (90 phút)

- 2.1 Hướng dẫn thực hành bài số 03
- 2.2 Hướng dẫn thực hành bài số 04

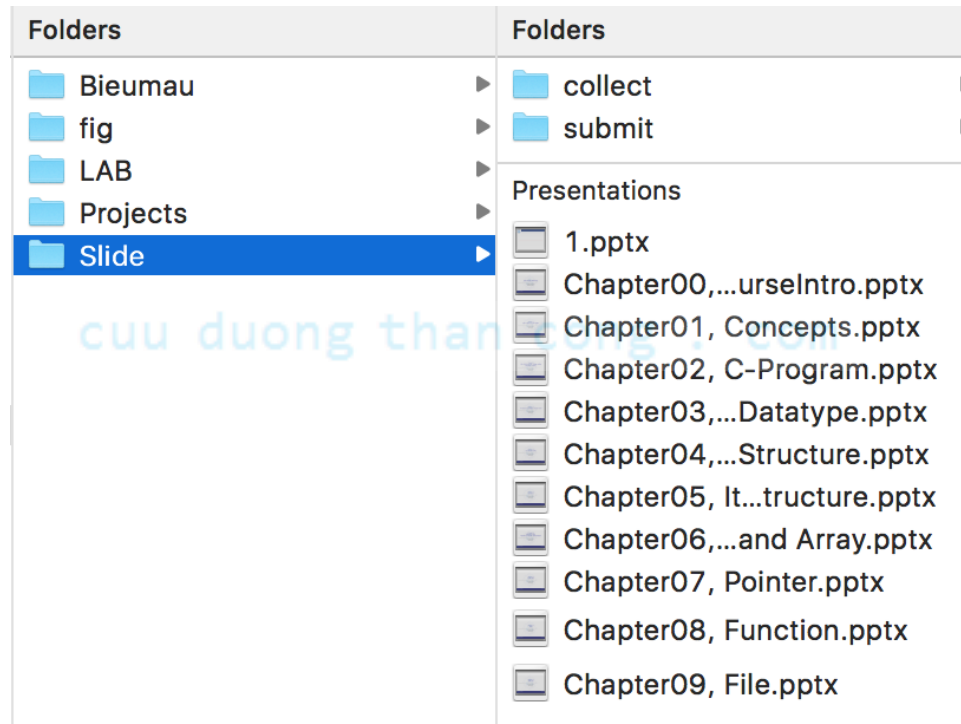
## III. Bài tập tự giải (90 phút)

# I. HƯỚNG DẪN BAN ĐẦU

- Sinh viên cần cài đặt được công cụ thực hành.
- Yêu cầu sinh viên đọc lại và ghi nhớ các kiến thức lý thuyết trong mục E trong tài liệu “*B14\_Tailieu\_TTLTCB*”
- Hướng dẫn chi tiết ví dụ mẫu để hiểu rõ về cách khai báo và thao tác trên tệp văn bản

# 1.1 TỔNG HỢP LÝ THUYẾT

- 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.



# 1.1 TỔNG HỢP LÝ THUYẾT

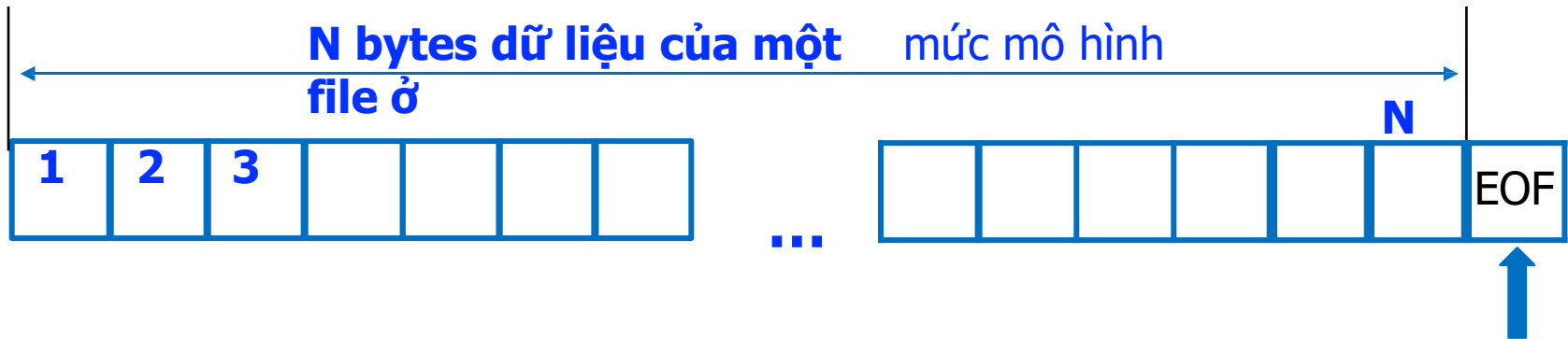
- Tập tin là một dãy các bytes dữ liệu, như hình vẽ, kết thúc bằng ký hiệu đặc biệt EOF
  - EOF (End Of File): là giá trị đặt biệt, không trùng với bất cứ giá trị của byte dữ liệu nào.
  - EOF: Ký hiệu mà các hàm đọc dữ liệu trả về để cho biết kết thúc tập tin.
  - (Nhiều hệ thống EOF = -1)





# 1.1 TỔNG HỢP LÝ THUYẾT

## 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)

# 1.1 TỔNG HỢP LÝ THUYẾT

- **Phân loại: 2 loại**

- Tập văn bản (text)
- Tập nhị phân (binary)

- **Tập văn bản:**

- Lưu trữ: các ký tự nhị phân được mã hóa dạng mã ASCII:  $8\text{bit} = 1\text{ byte} = 1\text{ ký tự}$ .
- 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.

- ❖ **Tập 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.

# 1.1 TỔNG HỢP LÝ THUYẾT

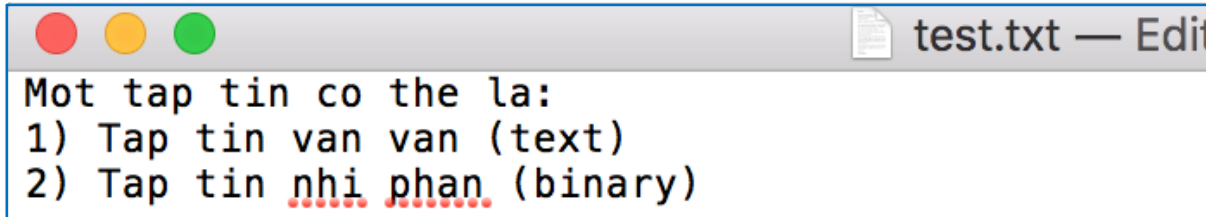
Trong C++, khi thao tác với một tệp dữ liệu, cần thực hiện tuần tự theo các bước như sau:

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

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

# 1.1 TỔNG HỢP LÝ THUYẾT

- Tập văn bản

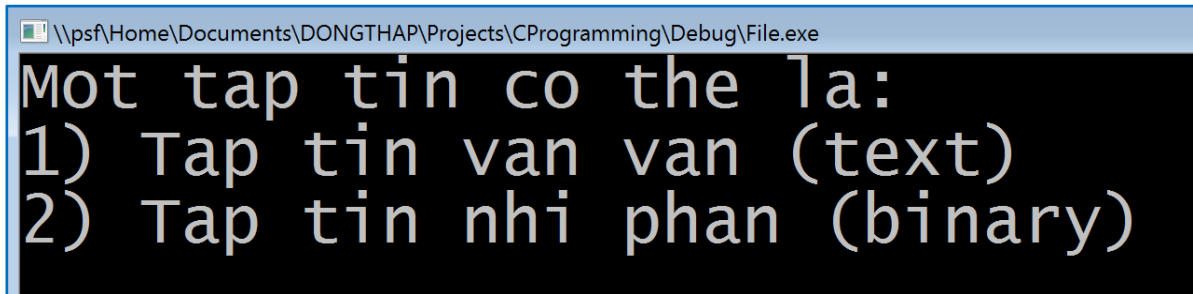


```
Mot tap tin co the la:  
1) Tap tin van van (text)  
2) Tap tin nhi phan (binary)
```

**Vào: tập tin văn bản (đọc được)**

**Chương trình**

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



```
Mot tap tin co the la:  
1) Tap tin van van (text)  
2) Tap tin nhi phan (binary)
```

**Ra: in lại các dòng trên màn hình**

# 1.1 TỔNG HỢP LÝ THUYẾT

## (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

# 1.1 TỔNG HỢP LÝ THUYẾT

## (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 bit đã được định nghĩa sẵn bởi C++.

# 1.1 TỔNG HỢP LÝ THUYẾT

## (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.

# 1.1 TỔNG HỢP LÝ THUYẾT

## (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 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.



# 1.1 TỔNG HỢP LÝ THUYẾT

- **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 nhị phân
  - Không nên mở tệp đồng thời với hai chế độ vừa ghi vừa đọc, nên tách thành 2 lần mở tệp riêng.

# 1.1 TỔNG HỢP LÝ THUYẾT

- Ví dụ:

*fstream*    *f* (“sohoc.txt”, *ios::in*);

*fstream*    *f1* (“D:\\Tep\\abc.txt”, *ios::in*);

Tên biến tệp



Tên tệp

Chế độ mở tệp

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

# 1.1 TỔNG HỢP LÝ THUYẾT

## (3) Thao tác với tệp tin

- Ghi 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 tệp tin trong vòng lặp:

- *while(!filevar.eof()) ...*

# 1.1 TỔNG HỢP LÝ THUYẾT

## (4) Đóng tệp

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

# 1.1 TỔNG HỢP LÝ THUYẾT

## (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

# 1.1 TỔNG HỢP LÝ THUYẾT

## (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 bit 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.

# 1.1 TỔNG HỢP LÝ THUYẾT

## (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>);*

# 1.1 TỔNG HỢP LÝ THUYẾT

- 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>);*



# 1.1 TỔNG HỢP LÝ THUYẾT

## (3) Thao tác với tệp tin

### 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 tệp tin trong vòng lặp:

- `while(!filevar.eof()) ...`

### (4). Đóng tệp

`filevar.close();`

# 1.1 TỔNG HỢP LÝ THUYẾT

---

- **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?

# 1.1 TỔNG HỢP LÝ THUYẾT

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

# 1.1 TỔNG HỢP LÝ THUYẾT

```
//Doc du lieu
```

```
f.open("a.dat", ios::in | ios::binary);
```

```
if(!f)
```

```
{cout <<"Khong the mo duoc tep tin "<<f<< endl;  
    exit(1);
```

```
}
```

```
f.read((char *) (b), sizeof(b)); //doc du lieu tu tep
```

```
for (int i = 0; i < SIZE; i++)
```

```
    cout << b[i] << "\\t";
```

```
cout << endl;
```

```
f.close();
```

```
return 0;
```

```
}    //end main
```

## 1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

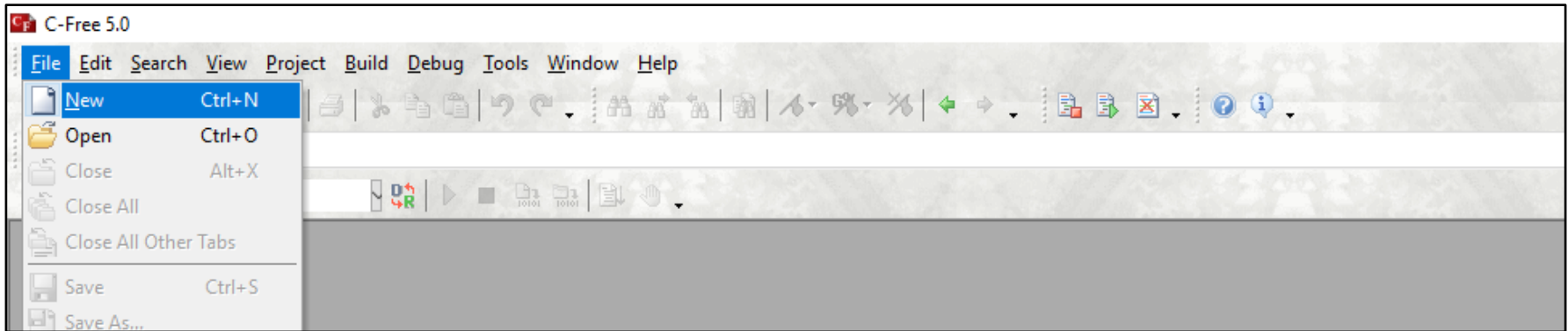
Trong bài thực hành này sinh viên tìm hiểu và thực hành cách ghi thông tin vào tệp văn bản.

### **Bài toán:**

Viết chương trình nhập vào một số nguyên dương. Kiểm tra số đó chẵn hay lẻ, kết quả lưu vào tệp văn bản “output.txt”

# 1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

- **Bước 1:** Tạo mới một file\*.cpp thực hiện thao tác File\New



- File mới xuất hiện, sinh viên chuyển sang bước 2 thực hiện gõ các câu lệnh theo các bước hướng dẫn.

## 1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

- **Bước 2:** Khai báo thư viện cần dùng

```
#include <iostream.h>
```

```
#include <fstream.h>
```

- **Bước 3:** Khai báo hàm main() là hàm chính của chương trình

```
int main()
```

- **Bước 4:** Khai báo biến nguyên.

```
int n;
```

## 1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

- **Bước 5:** Khai báo và mở tệp:

```
fstream f;  
f.open("output.txt",ios::out); //mo de ghi  
if (!f)  
{  
    cout<<"Khong tao duoc tep!"<<endl;  
    return -1;  
}
```



## 1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

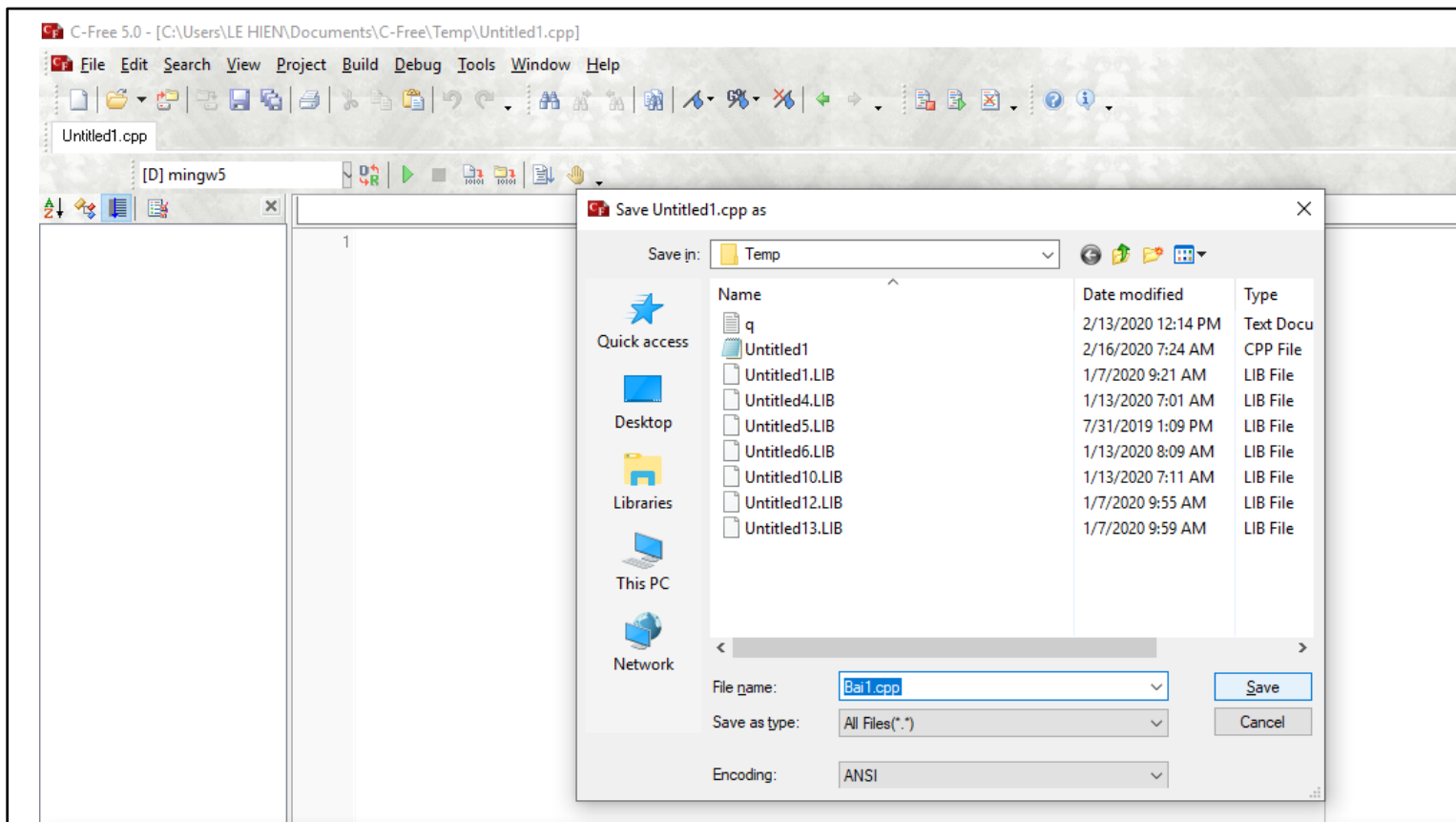
- **Bước 6:** Kiểm tra tính chất n và ghi tệp.

```
cout<<"Nhap n: "; cin>>n;  
if (n%2==0)  
    f<<n<<" la so chan"<<endl;  
else  
    f<<n<<" la so le"<<endl;
```

# 1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

- **Bước 7:** Lưu bài, chạy kiểm tra chương trình

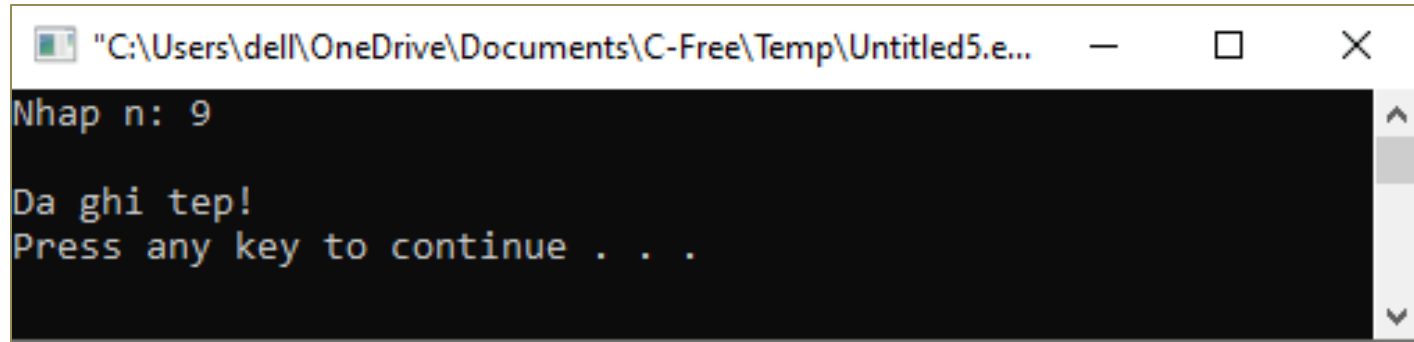
Sinh viên chọn trên thanh công cụ: File\Save\Gõ tên file cần lưu\Save



Ấn F5 để chạy và kiểm tra chương trình

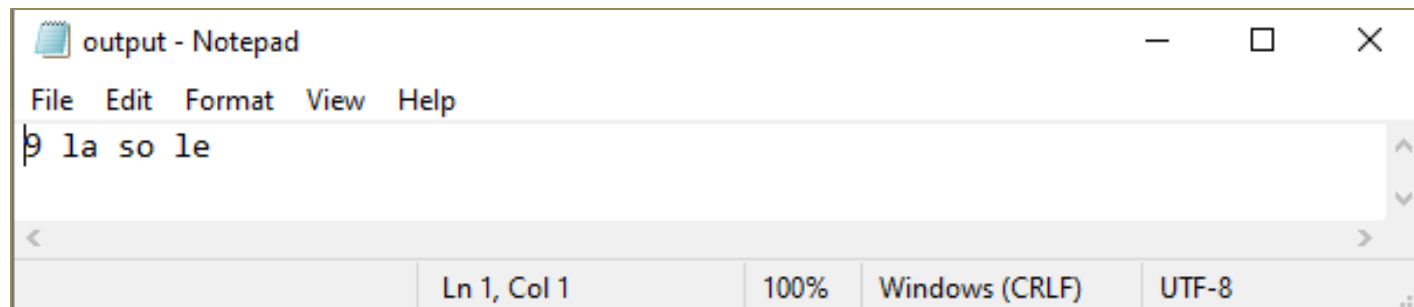
## 1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

Biên dịch chương trình, nếu có lỗi phải sửa lỗi, nếu chương trình chạy đúng sẽ cho kết quả như sau:



```
"C:\Users\del\OneDrive\Documents\C-Free\Temp\Untitled5.e...  
Nhap n: 9  
Da ghi tep!  
Press any key to continue . . .
```

- Tập output.txt



```
output - Notepad  
File Edit Format View Help  
Đ la so le  
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

**Tóm lại:** Trong bài thực hành 01 chúng ta đã biết cách ghi thông tin vào tệp văn bản.

## 1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

```
1. #include <iostream.h>
2. #include <fstream.h>
3. int main()
4. { int n;
5.   fstream f;
6.   f.open("output.txt", ios::out);
7.   if (!f)
8.   {   cout<<"Khong tao duoc tep!"<<endl;
9.       return -1;
10.  }
```

## 1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

```
11.  cout<<"Nhap n: "; cin>>n;
12.  if (n%2==0)
13.      f<<n<<" la so chan"<<endl;
14.  else
15.      f<<n<<" la so le"<<endl;
16.  cout<<"\nDa ghi tep!<<endl;
17.  return 0;
18. }
```

## 1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Trong bài thực hành này sinh viên tìm hiểu và thực hành thao tác đọc tệp văn bản

### **Bài toán:**

Cho trước tệp văn bản “data.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 “luu.txt”

## 1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

**Bước 1:** Khai báo thư viện cần dùng

```
#include <iostream.h>
```

```
#include <fstream.h>
```

## 1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

**Bước 2:** Hàm tìm UCLN của 2 số nguyên

```
int UCLN(int m, int n)
{
    while (m != n)
        if (m > n)
            m = m - n;
        else
            n = n - m;
    return m;
}
```



## 1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

**Bước 3:** Hàm đọc tệp, ghi dữ liệu vào 2 biến nguyên

```
void doctep(char tentep[], int &m, int &n)
{
    fstream f1(tentep, ios::in);
    if(!f1)
    {
        cout<<"Khong the mo duoc tep"<<f1<<endl;
        exit(1);
    }
    f1>>m>>n;
    f1.close();
}
```

## 1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

**Bước 4:** Hàm ghi tệp gọi hàm tìm UCLN của 2 số và lưu kết quả vào tệp

```
void ghitep(char tentep[], int &m, int &n)
{
    fstream f2(tentep, ios::out);
    if(!f2)
    { cout<<"Khong the tao duoc tep"<<f2<<endl;
      exit(1);
    }
    f2<<"\n"<<"UCLN = "<<UCLN(m,n) ;
}
```

## 1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

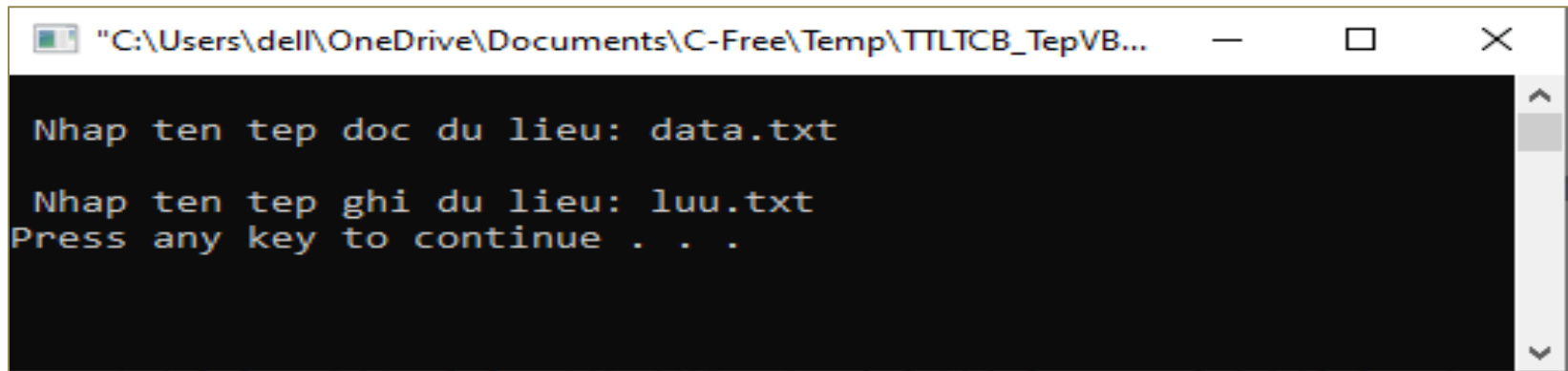
**Bước 5:** Khai báo hàm main() là hàm chính của chương trình

```
int main()
{ int      n, m;
  cout<<"\n Nhap ten tep doc du lieu: ";
  cin.getline(f, 30);
  doctep(f, m, n);
  cout<<"\n Nhap ten tep ghi du lieu: ";
  cin.getline(f, 30);
  ghitep(f, m, n);
  return 0;
}
```

## 1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

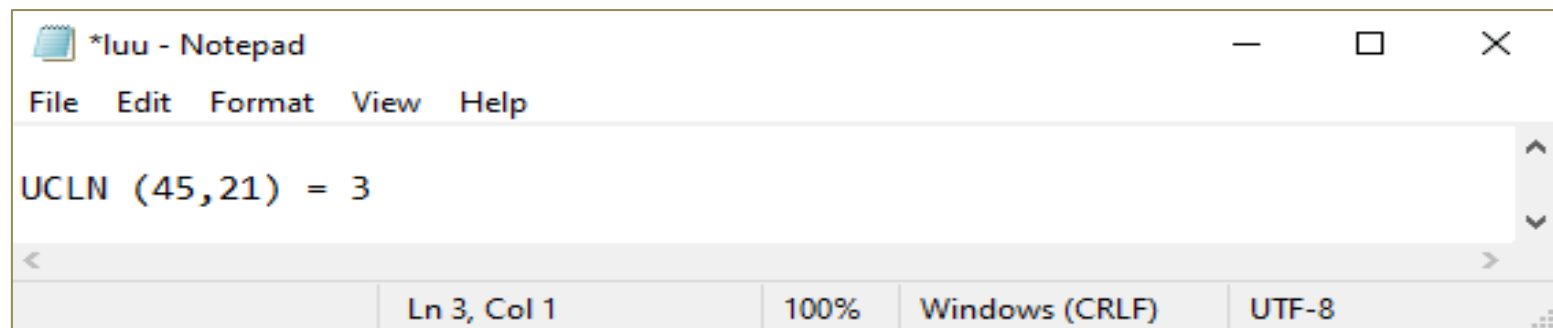
**Bước 6:** Thực hiện tương tự như bài thực hành 01

Biên dịch chương trình, nếu có lỗi phải sửa lỗi, nếu chương trình chạy đúng sẽ cho kết quả như sau:



```
"C:\Users\dell\OneDrive\Documents\C-Free\Temp\TTLTCB_TepVB...  
Nhap ten tep doc du lieu: data.txt  
Nhap ten tep ghi du lieu: luu.txt  
Press any key to continue . . .
```

Tệp luu.txt



```
*luu - Notepad  
File Edit Format View Help  
UCLN (45,21) = 3  
Ln 3, Col 1 100% Windows (CRLF) UTF-8
```

## 3.1 HƯỚNG DẪN THỰC HÀNH BÀI 05

```
1. #include<iostream.h>
2. #include<fstream.h>
3. int UCLN(int m, int n)
4. {   while (m!=n)
5.         if (m>n)
6.                 m=m-n;
7.         else
8.                 n=n-m;
9.         return m;
10. }
```

## 3.1 HƯỚNG DẪN THỰC HÀNH BÀI 05

```
11. void doctep(char tentep[], int &m, int &n)
12. {fstream f1(tentep, ios::in);
13.  if(!f1)
14.  {      cout<<"Khong mo duoc tep"<<f1<<endl;
15.        exit(1);      }
16.    f1>>m>>n;
17.    f1.close();
18. }
19. void ghitep(char tentep[], int &m, int &n)
20. {fstream f2(tentep, ios::out);
21.  if(!f2)
22.  {      cout<<"Khong tao duoc tep"<<f2<<endl;
23.        exit(1);      }
24.  f2<<"\n"<<"UCLN = "<<UCLN(m, n);
25. }
```

## 3.1 HƯỚNG DẪN THỰC HÀNH BÀI 05

```
26.int    main()  
27.{ int    n, m;  
28.    cout<<"\n Nhap ten tep doc du lieu: ";  
29.    cin.getline(f,30);  
30.    doctep(f, m, n);  
31.    cout<<"\n Nhap ten tep ghi du lieu: ";  
32.    cin.getline(f,30);  
33.    ghitep(f, m, n);  
34.    return 0;  
35.}
```

## II. HƯỚNG DẪN THƯỜNG XUYỀN

- Sinh viên tiếp tục thực hành các bài tập theo sự gợi ý hướng dẫn.

Trong phần này yêu cầu:

+ Một số phần trong bài sinh viên phải tự thực hiện (Ví dụ: tạo file ban đầu, khai báo thư viện, những hàm có tính chất tương tự ở ví dụ trước).

+ Sinh viên phải tự hoàn thiện chương trình và chạy đúng.



## 2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

- **Bài toán:**

Viết chương trình ghi các số chẵn trong phạm vi từ 1 đến 100 vào tệp văn bản “so\_chan.txt”?

## 2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

```
1. #include<iostream.h>           //Cach 1
2. #include<fstream.h>
3. void  GhiSoChan(ofstream &f1)
4. {
5.     int dem = 0;
6.     f1<<"Day so chan tu 1 -> 100 \n";
7.     for(int a = 1; a <= 100; a++)
8.     {
9.         if (a%2 == 0)
10.        {
11.            dem ++;
12.            f1<<a;
13.        }
```

## 2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

```
14.  if (dem % 10 == 0)
15.      f1<<"\n";
16.  if (dem % 10 != 0)
17.      f1<<"\t";
18.  }
19.  cout<<"Da ghi xong tep";
20. }
```

## 2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

```
21.int main()  
22.{ ofstream      f("so_chan.txt");  
23.  if (!f)  
24.    { cout<<"Khong the tao tep"<<endl;  
25.      return      -1;  
26.    }  
27.  GhiSoChan (f) ;  
28.  f.close () ;  
29.  return 0;  
30.}
```

## 2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

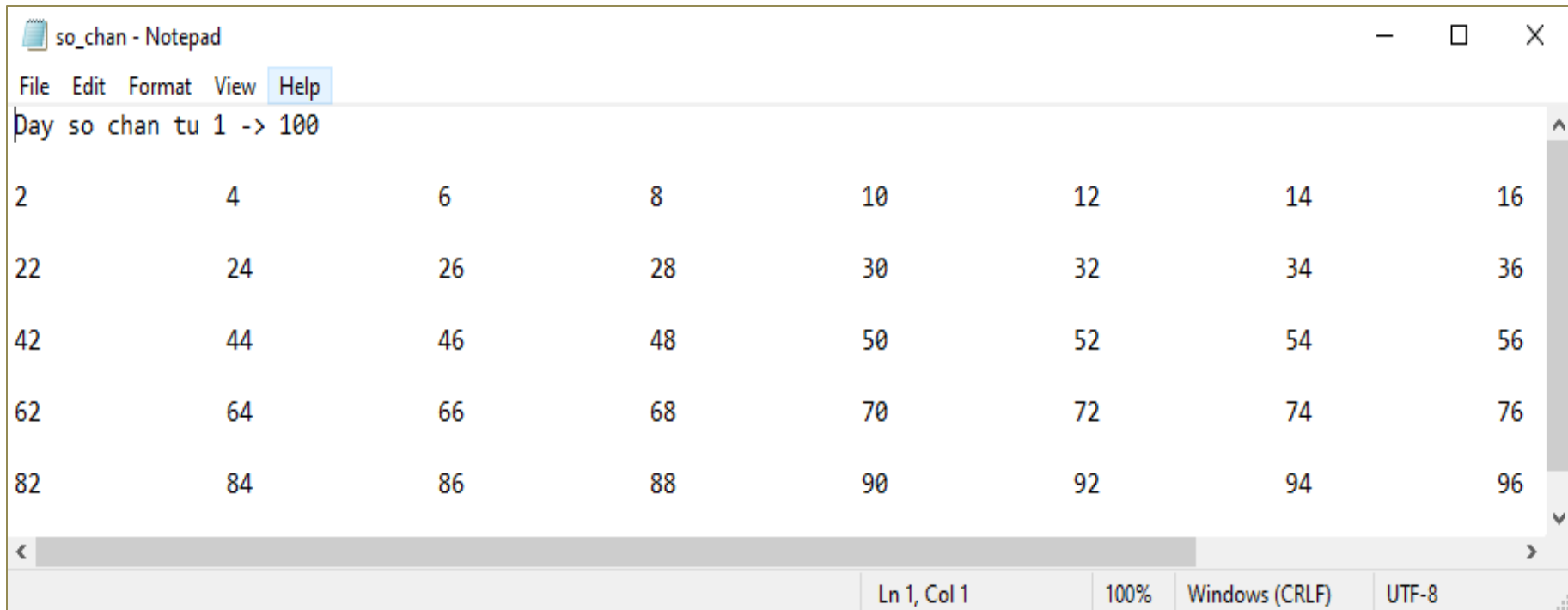
```
1. #include <iostream.h>           //Cach 2
2. #include <fstream.h>
3. int main()
4. { int dem =0;
5.   fstream f("so_chan.txt", ios::out);
6.   if(!f)
7.   {   cout<<"Khong the mo tep"<<endl;
8.       return -1;
9.   }
10.   f<<"Day so chan tu 1 -> 100 \n";
```

## 2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

```
11. for (int a =1; a<=100; a++)  
12. {      if (a%2 == 0)  
13.     {   dem++;      f<<a; }  
14.     if (dem %10 == 0)  
15.         f<<"\n";  
16.     if (dem%10 != 0)  
17.         f<<"\t";  
18. }  
19. f.close();  
20. return 0;  
21. }
```

## 2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

### Tệp sochan.txt:



```
so_chan - Notepad
File Edit Format View Help
Day so chan tu 1 -> 100

2      4      6      8      10     12     14     16
22     24     26     28     30     32     34     36
42     44     46     48     50     52     54     56
62     64     66     68     70     72     74     76
82     84     86     88     90     92     94     96

Ln 1, Col 1  100%  Windows (CRLF)  UTF-8
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

Viết chương trình quản lý lương nhân viên trong một công ty với thông tin của mỗi nhân viên bao gồm: {mã nv; họ tên; số đt; lương cơ bản; hệ số; lương tháng; phụ cấp; thu nhập}. Trong đó {mã nv; họ tên; số đt; lương cơ bản; hệ số; phụ cấp} được nhập vào từ bàn phím. Xây dựng chương trình thực hiện các chức năng sau đây:

1. Nhập danh sách nhân viên từ bàn phím
2. In danh sách nhân viên
3. Tính lương tháng của nhân viên với  $\text{lương tháng} = \text{lương cơ bản} * \text{hệ số}$ .
4. Tính thu nhập của nhân viên với  $\text{thu nhập} = \text{lương tháng} + \text{phụ cấp}$
5. Sắp xếp danh sách nhân viên theo thứ tự tăng dần về thu nhập
6. In thông tin nhân viên có thu nhập cao nhất
7. In thông tin tổng lương phải trả cho nhân viên của công ty
8. In danh sách các nhân viên có hệ số lương  $> \text{hs}$  với  $\text{hs}$  là một số thực được nhập vào từ bàn phím
9. Lưu danh sách nhân viên ra tệp nv.dat
10. Đọc danh sách nhân viên từ tệp nv.dat và kết xuất thông tin lên màn hình



## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
#include <iostream.h>
#include <iomanip.h>
#include <fstream.h>
typedef struct
{
    char        manv[10];
    char        ten[30];
    char        sdt[12];
    float        hesoluong;
    float        luongthang;
    int         luongcb;
    float        phucap;
    double       thunhap;
} nhanvien;
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
void nhap_lnv(nhanvien *x)
{
    cin.ignore();
    cout<<"Nhap ma nhan vien: ";
    cin.getline(x->manv,10);
    cout<<"Nhap ten nhan vien: ";
    cin.getline(x->ten, 30);
    cout<<"Nhap so dien thoai nhan vien: ";
    cin.getline(x->sdt, 12);
    cout<<"Nhap luong co ban nhan vien: ";
    cin>>x->luongcb;
    cout<<"Nhap phu cap cua nhan vien: ";
    cin>>x->phuicap;
    cout<<"Nhap he so luong cua nhan vien: ";
    cin>>x->hesoluong;
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
void nhap_ds( nhanvien a[], int n)
{
    for(int i=0; i<n; i++)
    {
        cout<<"Nhap thong tin nhan vien thu
: "<<i+1<<endl;
        nhap_lnv(&a[i]);
    }
}
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
void    tieude ()
{
    cout<<setw(15)<<"Ma nhan vien"<<"|";
    cout<<setw(20)<<"Ten nhan vien"<<"|";
    cout<<setw(15)<<"So dien thoai"<<"|";
    cout<<setw(15)<<"He so luong"<<"|";
    cout<<setw(15)<<"Luong thang"<<"|";
    cout<<setw(15)<<"Luong Co Ban"<<"|";
    cout<<setw(15)<<"Phu Cap"<<"|";
    cout<<setw(15)<<"Thu nhap"<<"| "<<endl;
}
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
void xuat_nv(nhanvien a[], int n)
{cout<<"----- DANH SACH NHAN VIEN --
-----"<<endl;
cout<<"-----
-----
-----
-----"<<endl;
tieude();
for (int i=0; i<n; i++)
{   cout<<setw(15)<<a[i].manv<<" | ";
    cout<<setw(20)<<a[i].ten<<" | ";
    cout<<setw(15)<<a[i].sdt<<" | ";
    cout<<setw(15)<<a[i].hesoluong<<" | ";
    cout<<setw(15)<<a[i].luongthang<<" | ";
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
cout<<setw(15)<<a[i].luongcb<<"|";  
cout<<setw(15)<<a[i].phucap<<"|";  
cout<<setw(15)<<a[i].thunhap<<"| "<<endl;  
}  
cout<<"-----  
-----  
-----  
-----"<<endl;  
}
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
void tingh_luong_thang(nhanvien a[], int n)
{
    for(int i=0; i<n; i++)
        a[i].luongthang =
        a[i].luongcb*a[i].hesoluong ;
}

void tingh_thu_nhap(nhanvien a[], int n)
{
    for(int i=0; i<n; i++)
        a[i].thunhap =
        (double)a[i].luongcb+a[i].phucap;
}
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
void tntangdan(nhanvien a[], int n)
{
    for (int i=0; i<n-1; i++)
        for (int j=i+1; j<n; j++)
            if(a[i].thunhap > a[j].thunhap)
            {
                nhanvien tg = a[i];
                a[i] = a[j];
                a[j] = tg;
            }
    xuat_nv(a,n) ;
}
```



## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
int nhan_vien_luong_cao_nhat(nhanvien a[],int
n)
{
    float max=a[0].thunhap;
    int vitri=0;
    for(int i=1;i<n;i++)
        if(max < a[i].thunhap)
        {
            max = a[i].thunhap;
            vitri =i;
        }
    // bang ten nhan vien co thu nhap cao nhat
    cout<<"DANH SACH NHAN THU NHAP CAO NHAT"<<endl;
    tieude();
}
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
cout<<"-----"<<endl;
for(int i=0;i<n;i++)
    if(a[i].thunhap == max)
    {cout<<setw(15)<<a[i].manv<<"|";
      cout<<setw(20)<<a[i].ten<<"|";
      cout<<setw(15)<<a[i].sdt<<"|";
      cout<<setw(15)<<a[i].hesoluong<<"|";
      cout<<setw(15)<<a[i].luongthang<<"|";
      cout<<setw(15)<<a[i].luongcb<<"|";
      cout<<setw(15)<<a[i].phucap<<"|";
      cout<<setw(15)<<a[i].thunhap<<"| "<<endl;
    }
cout<<"-----"<<endl;
}
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
void xuat_theo_hs(nhanvien a[], int n, float
hs)
{cout<<"----- DANH SACH NHAN VIEN -----"<<endl;
cout<<"-----"<<endl;
tieude();
for (int i=0; i<n; i++)
    if(a[i].hesoluong > hs )
    {
        cout<<setw(15)<<a[i].manv<<" | ";
        cout<<setw(20)<<a[i].ten<<" | ";
        cout<<setw(15)<<a[i].sdt<<" | ";
        cout<<setw(15)<<a[i].hesoluong<<" | ";
        cout<<setw(15)<<a[i].luongthang<<" | ";
        cout<<setw(15)<<a[i].luongcb<<" | ";
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
    cout<<setw(15)<<a[i].phucap<<" | ";  
    cout<<setw(15)<<a[i].thunhap<<" | "<<endl;  
}  
cout<<"-----"<<endl;  
}
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
int main()
{nhanvien ql[100], nv[100];
  int n;          //n la so nha vien
  float hs;
  cout<<"Nhap vao so nhan vien: ";cin>>n;
  nhap_ds(ql,n);
  tinh_luong_thang(ql,n);
  tinh_thu_nhap (ql,n);
  xuat_nv(ql,n);
  cout<<"Nhap he so mong muon :";cin>>hs;
  xuat_theo_hs(ql,n,hs);
  cout<<"Bang luong xep theo thu tu tang dan"<<endl;
  tntangdan (ql,n);
  nhan_vien_luong_cao_nhat (ql,n);
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
// Doc du lieu
f.open("nv.dat", ios::in | ios::binary);
if(!f)
{ cout<<"Khong the mo duoc tep"<<f<< endl;
  exit(1);
}
f.read((char *) (nv), sizeof(nv));
//in mang nv ra man hinh
xuat_nv(nv,n);
cout << endl;
f.close();
return 0;
}
```

## 2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

- Kết quả khi chạy chương trình

```
"C:\Users\dell\OneDrive\Documents\C-Free\Temp\Untitled12.exe"
Nhap ma nhan vien: 123
Nhap ten nhan vien: Thanh
Nhap so dien thoai nhan vien: 0988919368
Nhap luong co ban nhan vien: 17500
Nhap phu cap cua nhan vien: 50000
Nhap he so luong cua nhan vien: 3.3
Nhap thong tin nhan vien thu : 2
Nhap ma nhan vien: 456
Nhap ten nhan vien: Hang
Nhap so dien thoai nhan vien: 0974161312
Nhap luong co ban nhan vien: 17500
Nhap phu cap cua nhan vien: 75000
Nhap he so luong cua nhan vien: 3.6
----- DANH SACH NHAN VIEN -----
Ma nhan vien| Ten nhan vien| So dien thoai| He so luong| Luong thang| Luong Co Ban| Phu Cap| Thu nhap|
123| Thanh| 0988919368| 3.3| 57750| 17500| 50000| 67500|
456| Hang| 0974161312| 3.6| 63000| 17500| 75000| 92500|
-----
Nhap he so mong muon :3.0
----- DANH SACH NHAN VIEN -----
Ma nhan vien| Ten nhan vien| So dien thoai| He so luong| Luong thang| Luong Co Ban| Phu Cap| Thu nhap|
123| Thanh| 0988919368| 3.3| 57750| 17500| 50000| 67500|
456| Hang| 0974161312| 3.6| 63000| 17500| 75000| 92500|
-----
Bang luong xep theo thu tu tang dan
----- DANH SACH NHAN VIEN -----
Ma nhan vien| Ten nhan vien| So dien thoai| He so luong| Luong thang| Luong Co Ban| Phu Cap| Thu nhap|
123| Thanh| 0988919368| 3.3| 57750| 17500| 50000| 67500|
456| Hang| 0974161312| 3.6| 63000| 17500| 75000| 92500|
-----
----- DANH SACH NHAN THU NHAP CAO NHAT -----
Ma nhan vien| Ten nhan vien| So dien thoai| He so luong| Luong thang| Luong Co Ban| Phu Cap| Thu nhap|
456| Hang| 0974161312| 3.6| 63000| 17500| 75000| 92500|
-----
----- DANH SACH NHAN VIEN -----
Ma nhan vien| Ten nhan vien| So dien thoai| He so luong| Luong thang| Luong Co Ban| Phu Cap| Thu nhap|
123| Thanh| 0988919368| 3.3| 57750| 17500| 50000| 67500|
456| Hang| 0974161312| 3.6| 63000| 17500| 75000| 92500|
```

### III BÀI TẬP TỰ GIẢI

**Bài 5:** Viết chương trình ghi thông tin các số lẻ trong phạm vi từ 1 đến 100 vào tệp văn bản “so\_le.txt”?

**Bài 6:** Cho tệp văn bản “in6.txt” chứa ba số nguyên dương. Viết hàm đọc thông tin từ tệp, tìm số lớn nhất, nhỏ nhất. Kết quả ghi vào tệp văn bản “out6.txt”?

**Bài 7:** Cho tệp văn bản “in7.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 vào tệp văn bản “out7.txt”?

**Bài 8:** Cho tệp văn bản “in8.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 “out8.txt”?



### III BÀI TẬP TỰ GIẢI

**Bài 9:** Cho tệp văn bản “in9.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:

- Tìm số nguyên tố, hoàn hảo, số đối xứng. Kết quả ghi vào cuối tệp văn bản “in9.txt”
- Sắp xếp mảng tăng dần sử dụng Interchange sort? Kết quả ghi vào tệp văn bản “out9.txt”

### III BÀI TẬP TỰ GIẢI

*Ví dụ: Ban đầu tệp “in9.txt” chứa:*

*5*

*28 17 70 31 11*

*Sau khi thực hiện chương trình tệp “in9.txt”*

*5*

*28 17 70 31 11*

*So nguyên tố là : 17 11 31*

*So hoán hao là: 28*

*So đối xứng là: 11*

### III BÀI TẬP TỰ GIẢI

**Bài 10:** Cho tệp văn bản “in10.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:

- 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 “in5.txt”
- Sắp xếp mảng giảm dần sử dụng selection sort? Kết quả ghi vào tệp văn bản “out10.txt”

### III BÀI TẬP TỰ GIẢI

*Ví dụ: Ban đầu tệp “in10.txt” chứa:*

*5*

*21 30 5 30 10*

*Sau khi thực hiện chương trình tệp “in10.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 nhất :2*

### III BÀI TẬP TỰ GIẢI

**Bài 11:** 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”

### III BÀI TẬP TỰ GIẢI

**Bài 12:** Trong đó {mã book; tên book; tác giả; nhà XB; năm XB; đơn giá; số lượng; thuế VAT} được nhập vào từ bàn phím. Xây dựng menu thực hiện các chức năng sau đây:

- Nhập thông tin các cuốn sách từ bàn phím
- Tính cột thành tiền với  $\text{thành tiền} = \text{đơn giá} * \text{số lượng}$ .
- Tính cột thực tiền với  $\text{thực tiền} = \text{thành tiền} + \text{thuế VAT}$ .
- In danh sách thông tin các cuốn sách
- Sắp xếp các cuốn sách theo thứ tự giảm dần về năm XB. In ds sau sx
- In thông tin cuốn sách có thực tiền lớn nhất

### III BÀI TẬP TỰ GIẢI

- Thêm vào danh sách một cuốn sách với thông tin nhập vào từ bàn phím sao cho vẫn đảm bảo thứ tự sắp xếp
- Xóa khỏi danh sách các cuốn sách có thuế VAT > hs với hs là một số thực được nhập vào từ bàn phím
- Lưu danh sách các cuốn sách ra tệp books.dat
- Đọc danh sách các cuốn sách từ tệp books.dat và xuất thông tin ra màn hình

## 3.1 HƯỚNG DẪN THỰC HÀNH BÀI 05

**Bài 5:** Viết chương trình ghi thông tin các số lẻ trong phạm vi từ 1 đến 100 vào tệp văn bản “so\_le.txt”?



## 3.1 HƯỚNG DẪN THỰC HÀNH BÀI 05

```
#include<iostream.h>
#include<fstream.h>
void GhiSoLe(ofstream &file)
{
    int dem = 0;
    file<<"Day so le tu 1 -> 100 \n";
    for(int a = 1; a <= 100; a++)
    {
        if(a%2 != 0)
        {
            dem ++;
            file<<a;
        }
    }
}
```

## 3.1 HƯỚNG DẪN THỰC HÀNH BÀI 05

```
if (dem % 10 == 0)
    file<<"\n";
if (dem % 10 != 0)
    file<<"\t";
}
cout<<"Da ghi tep"<<endl;;
}
```

## 3.1 HƯỚNG DẪN THỰC HÀNH BÀI 05

```
int main()  
{ ofstream    f("so_le.txt");  
    if (!f)  
        { cout<<" Khong the mo tep";  
          return    -1;  
        }  
    GhiSoLe(f);  
    f.close();  
    return 0;  
}
```

## 3.2 HƯỚNG DẪN THỰC HÀNH BÀI 06

**Bài 6:** Cho tệp văn bản “in6.txt” chứa ba số nguyên dương. Viết hàm đọc thông tin từ tệp, tìm số lớn nhất, nhỏ nhất. Kết quả ghi vào tệp văn bản “out6.txt”?

## 3.1 HƯỚNG DẪN THỰC HÀNH BÀI 05

```
#include <iostream.h>
#include <fstream.h>
void doctep(char tentep[], int &x, int
&y, int &z)
{   fstream   f(tentep,ios::in);
    if(!f)
        { cout << "Khong the mo duoc tep tin
"<<f<<endl;
            exit(1);
        }
    f>>x>>y>>z;
    f.close();
}
```

## 3.1 HƯỚNG DẪN THỰC HÀNH BÀI 05

```
int max(int a, int b, int c)
{
    int m = a;
    if (m < b) m = b;
    if (m < c) m = c;
    return m;
}

int min(int a, int b, int c)
{
    int m = a;
    if (m > b) m = b;
    if (m > c) m = c;
    return m;
}
```

## 3.1 HƯỚNG DẪN THỰC HÀNH BÀI 05

```
void ghitep(char tentep[], int &x, int &y,
int &z)
{ fstream f1(tentep,ios::out);
  if (!f1)
  { cout<<"Khong tao duoc tep"<<f1<<endl;

      exit(1);
  }
  f1<<"\n"<<"So lon nhat trong 3 so la:
"<<max(x, y, z); //ghi vao tep
  f1<<"\n"<<"So nho nhat trong 3 so la:
"<<min(x, y, z); //ghi vao tep
  cout<<" Da ghi xong tep!"<<endl;
}
```

## 3.1 HƯỚNG DẪN THỰC HÀNH BÀI 05

```
int main()
{
    char        f[30];
    int         a, b, c;
    cout<<" Nhap ten tep doc du lieu:";
    cin.getline(f,30);
    doctep(f, a, b, c);
    cout<<"\n Nhap ten tep ghi du lieu:";
    cin.getline(f,30);
    ghitep(f, a, b, c);
    return 0;
}
```



## 3.4 HƯỚNG DẪN THỰC HÀNH BÀI 08

**Bài 8:** Cho sẵn tệp "dulieu.txt" chứa 3 số nguyên. Hãy đọc thông tin từ tệp. Kiểm tra xem 3 số có tạo thành 3 cạnh tam giác không. Kết quả lưu vào cuối tệp "dulieu.txt"?

## 3.4 HƯỚNG DẪN THỰC HÀNH BÀI 08

```
#include<iostream.h>
#include<fstream.h>

void doctep(int &x, int &y, int &z)
{
    fstream  f("dulieu.txt",ios::in);

    f>>x>>y>>z;
    f.close();
}
```

## 3.4 HƯỚNG DẪN THỰC HÀNH BÀI 08

```
int main()
{ int  a, b, c;
doctep(a,b,c);
fstream  f("dulieu.txt",ios::app);
if ((a+b>c) && (b+c>a) && (a+c>b) )
{      f<<"Day la 3 canh tam giac"<<endl;
      if(a==b &&b==c)
          f<<"Do la tam giac deu"<<endl;
      else
          if(a==b || b==c || c==a)
              f<<"Do la tam giac can"<<endl;
```

## 3.4 HƯỚNG DẪN THỰC HÀNH BÀI 08

```
else
    if(a*a + b*b==c*c || b*b + c*c == a*a ||
    a*a + c*c == b*b)
        f<<"Do la tam giac vuong"<<endl;
    else
        f<<"Do la tam giac thuong"<<endl;
}
else
    f<<"Khong tao thanh tam giac"<<endl;
f.close();
return 0;
}
```

## IV. TỔNG KẾT

**Kiến thức cần ghi nhớ trong buổi thực hành:**

1

Khái niệm tệp

2

Vào ra với tệp văn bản

3

Thao tác với con trỏ tệp

## V. GIAO NHIỆM VỤ TUẦN TIẾP THEO

1. Hoàn thành tất cả các bài thực hành trong buổi học.
2. Sinh viên đọc tài liệu và chuẩn bị trước nội dung học của tuần tiếp theo:

### **Giải bài tập với tệp nhị phân**

- Thao tác ghi tệp
- Thao tác đọc tệp.
- Phát hiện và sửa lỗi