

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN

Giáo trình thực hành
NHẬP MÔN
LẬP TRÌNH

Họ và tên Ngày sinh.....
Mã số sinh viên Lớp.....
Bậc Khoa

Điện thoại: Email.....

LƯU HÀNH NỘI BỘ
Năm 2020

MỤC LỤC

Bài 1: Làm quen với môi trường Visual C++ 2013	1
Bài 2: Các cấu trúc điều khiển	
Bài 3: Các cấu trúc lặp	
Bài 4: Cấu trúc lặp do...while.....	
Bài 5: Hàm – cơ chế truyền tham trị	
Bài 6: Hàm – cơ chế truyền tham biến.....	
Bài 7: Mảng một chiều	
Bài 8: Mảng hai chiều	
Bài 9: Xâu kí tự	
Bài 10: Ôn tập	

BÀI 1: LÀM QUEN VỚI MÔI TRƯỜNG VISUAL C++ 2017

I. MỤC TIÊU:

Sau khi thực hành xong bài này, sinh viên phải biết cách:

- Sử dụng đối tượng **cin** và **cout**.
- Biên dịch, sửa lỗi cú pháp, dịch ra tập tin thực thi và thực thi chương trình.
- Viết được một chương trình C++ đơn giản chỉ sử dụng lệnh nhập/xuất.

II. TÓM TẮT LÝ THUYẾT:

- **Cấu trúc một chương trình C++:**

```
#include <iostream>
using namespace std;
// khai báo các thư viện cần sử dụng các đối tượng
trong chương trình.
int main() { //Điểm bắt đầu của hàm main
    //nội dung chương trình chính
    cout << "Hello, there!";
    return 0;
} //Điểm kết thúc của hàm main
```

- **Đối tượng cout:**

Sử dụng ký hiệu << để gửi thông tin cho đối tượng **cout**.

Ví dụ:

```
cout << "Hello, there!"; //xuất ra màn hình nội dung
"Hello, there!"
```

▪ Đối tượng cin:

Sử dụng ký hiệu >> để gửi thông tin cho đối tượng cin;

Thông tin đưa vào dùng đối tượng cin đi sau ký hiệu >>.

Ví dụ:

```
int height;//khai báo biến height kiểu số nguyên
cout<<"Nha cao bao nhieu tang ?";
cin>>height;//nhập giá trị cho biến height
```

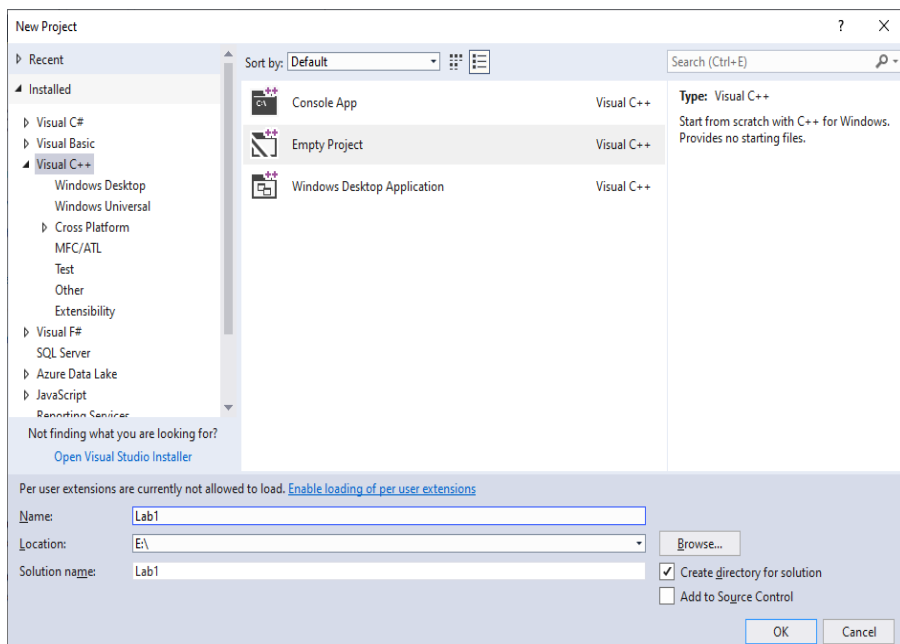
III. NỘI DUNG THỰC HÀNH:

1. Hướng dẫn sử dụng nhanh visual studio 2017

a) Các bước tạo một project đơn giản

- Bước 1: Tạo một project đơn giản

- Chọn File → New → Project

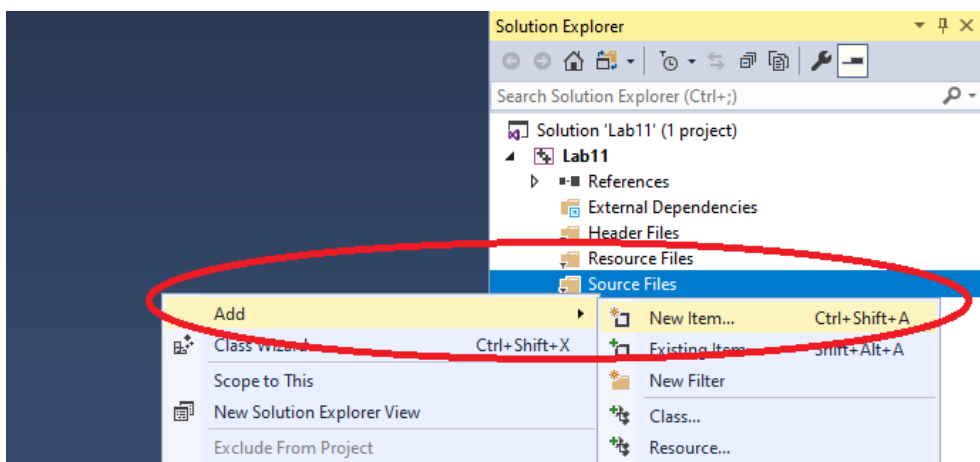


Bài 1: LÀM QUEN VỚI MÔI TRƯỜNG VISUAL C++ 2017

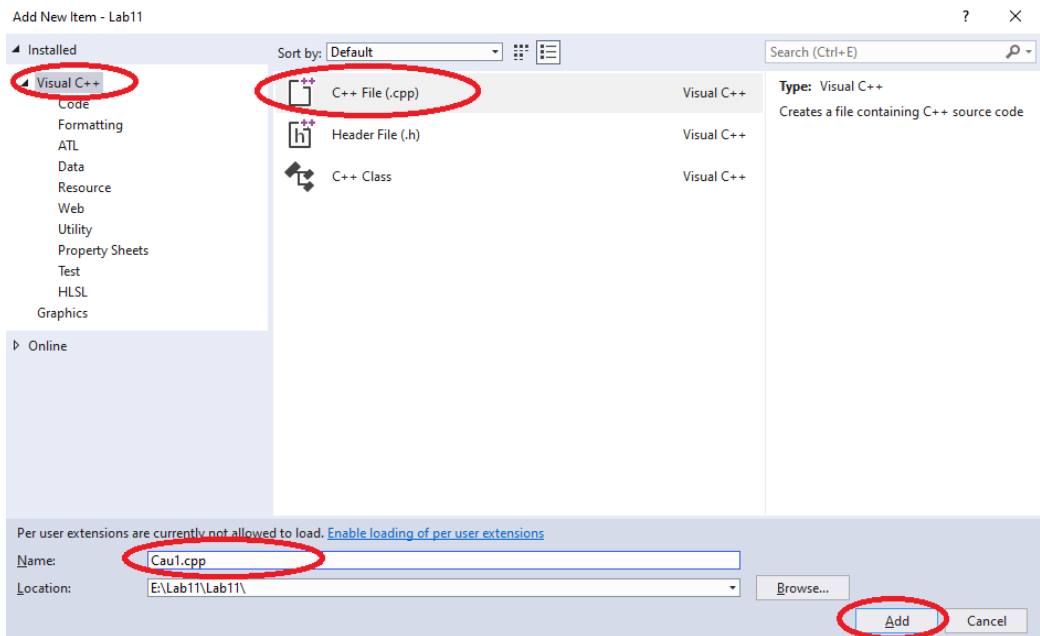
- Chọn Visual C++ → chọn Win32 → chọn Win32 Console Application
- Gõ tên của project muốn tạo vào mục **Name**; chọn thư mục cần lưu trữ trong mục **Location**. Toàn bộ các tập tin của project này đều nằm trong thư mục đã chỉ ra → **Ok**; một hộp thoại khác mở ra.

- Bước 2: Tạo một tập tin nguồn Bai1.cpp

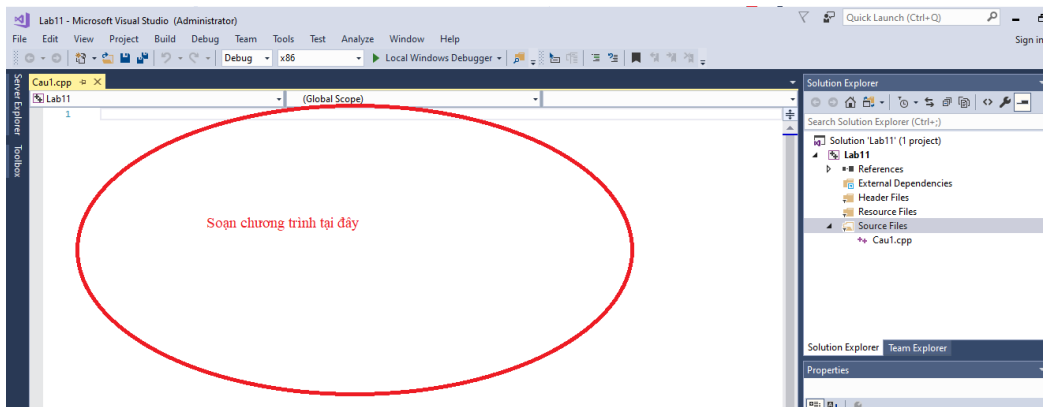
- Click chuột phải trên Source Files → chọn **Add** → chọn **New Item**.



- Một hộp thoại xuất hiện; tại hộp thoại này chọn Code → C++ File (.cpp) → đặt tên tập tin tại mục **Name** → nhấp chọn **Add**.



- **Bắt đầu soạn thảo nội dung tập tin .cpp tại cửa sổ này:**



- Bước 3: Biên dịch và thực thi chương trình

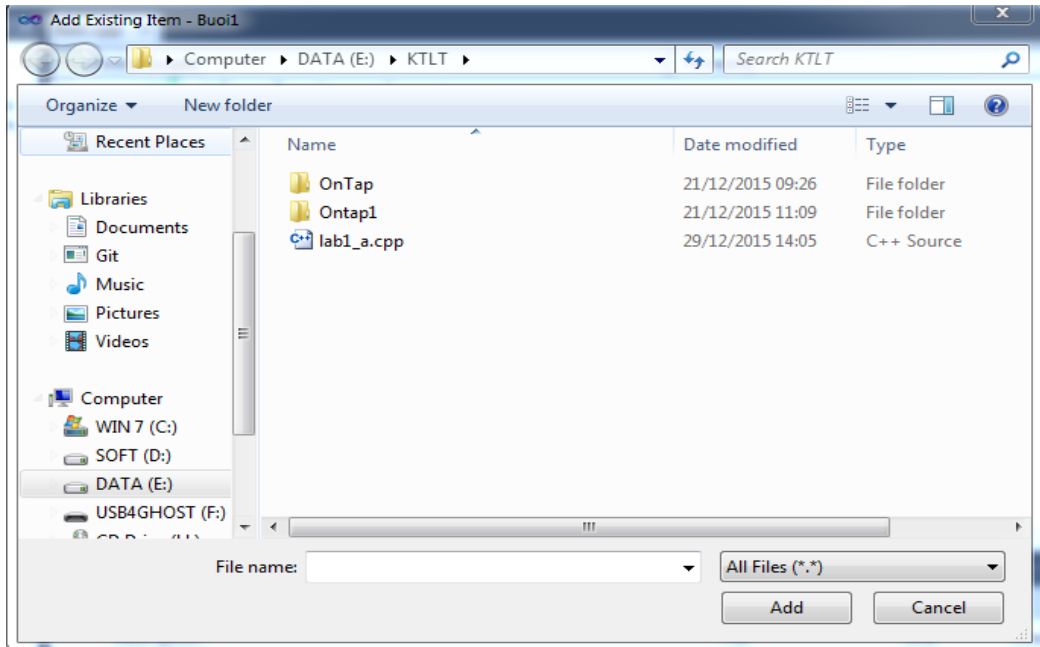
- Vào **Build** → chọn **Build Solution** để biên dịch chương trình (**F7**);
(tập tin .exe được tạo ra theo nguyên tắc là *ten_project.exe*)
- Vào **Build** → chọn **Compile Solution** để dịch chương trình, nếu có lỗi thì sửa lỗi đầu tiên và biên dịch lại. (**Ctrl + F7**)
- Vào **Debug** → chọn **Start Debugging** để chạy debug chương trình (chạy từng bước).

Bài 1: LÀM QUEN VỚI MÔI TRƯỜNG VISUAL C++ 2017

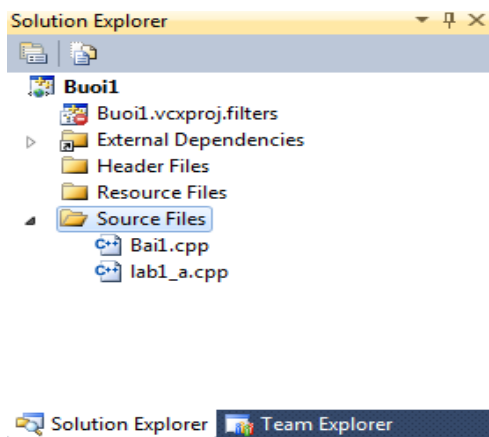
- Vào Debug → chọn Start without Debugging để thực thi chương trình.

b) Cách sử dụng lại một tập tin nguồn có sẵn.

- Click chuột phải trên Source Files → chọn Add → chọn Add Existing Item.



- Chọn tập tin được thêm vào project → chọn Add → tập tin được thêm vào project.



- Lưu ý:

- Tập tin được **add** vào vẫn được lưu tại vị trí ban đầu, không phải đã được đưa vào cùng vị trí của solution.
- Một project có thể có **nhiều tập tin .cpp** nhưng chỉ có **một hàm main()**.

c) Cách sửa lỗi chương trình.

- Khi biên dịch chương trình, nếu chương trình có lỗi, nhấn **F4** để về dòng lệnh kế sau dòng lệnh gây ra lỗi đầu tiên, sửa lỗi và biên dịch lại chương trình. Nếu có **nhiều lỗi** xảy ra, **chỉ sửa lỗi đầu tiên và dịch lại chương trình.**

d) Các phím tắt thường sử dụng

Phím	Ý nghĩa
Ctrl + S	Lưu file lần đầu tiên
Ctrl + O	Mở một file đã tồn tại
Ctrl + F7	Biên dịch lỗi
F7	Dịch thành file thực thi (*.exe)
Ctrl + F5	Thực thi chương trình
Alt + F4	Đóng cửa sổ đang mở.
F9	Tắt/mở dòng lệnh debug (breakpoint)
F5	Thực thi tại dòng lệnh chỉ định. (breakpoint)
Ctrl + F10	Thực thi tại dòng lệnh đang nhận con trỏ chuột
F10	Thực thi từng dòng lệnh
F11	Vào khối lệnh chỉ định

Shift + F11	Thoát ra khỏi khối lệnh chỉ định.
-------------	-----------------------------------

e) *Hướng dẫn debug chương trình*

e.1/ Thực thi từng dòng lệnh

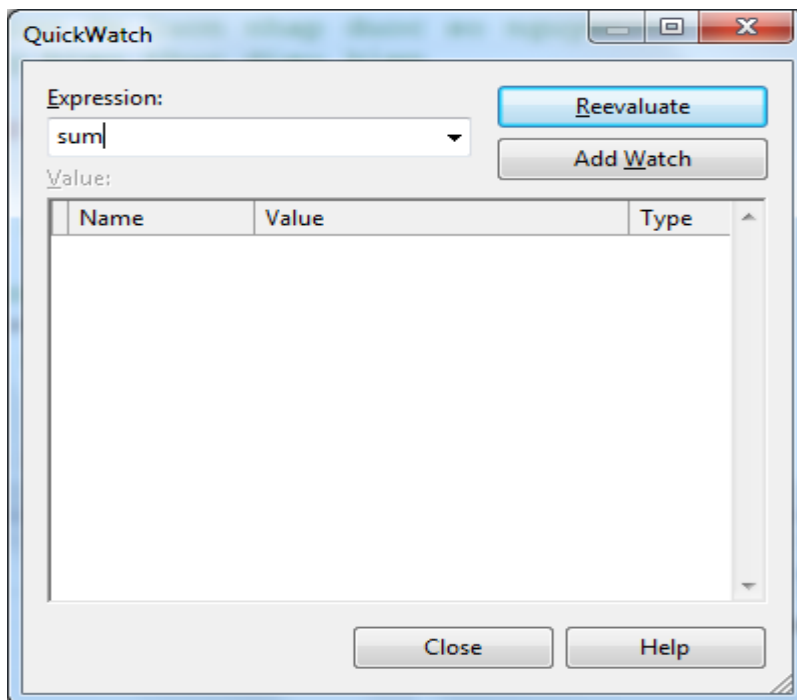
- Để thực thi từng dòng lệnh bắt đầu từ dòng lệnh đầu tiên, nhấn F10.
- Tiếp tục nhấn F10 để thực thi dòng lệnh kế tiếp.
- Muốn vào khối lệnh chỉ định, nhấn F11.
- Muốn thực thi từng bước tại dòng lệnh đang nhận con trỏ, nhấn Ctrl + F10.

e.2/ Thực thi **debug** tại các dòng lệnh chỉ định

- Thiết lập **breakpoint** tại các dòng lệnh chỉ định: đặt con trỏ tại dòng lệnh cần đặt breakpoint, nhấn **F9** (nhấn F9 lần nữa để bỏ breakpoint)
- Sau khi thiết lập breakpoint, nhấn **F5**.

e.3/ Kiểm tra giá trị các biến cần theo dõi khi **debug** chương trình

Nhấn tổ hợp phím **Shift + F9** để mở cửa sổ sau:



Muốn theo dõi hoạt động của biến nào, nhập vào tên biến ở khung Expression (ví dụ: sum) → chọn **Add Watch**.

2. Những quy tắc cơ bản khi soạn code

a) **Đặt tên biến:** tuân theo quy tắc đặt tên biến của C++, nên dùng kí tự viết thường cho từ đầu tiên, từ sau viết Hoa, có thể mô tả 1 phần kiểu dữ liệu của biến; thường gắn liền với ý nghĩa của biến sử dụng.

Ví dụ: `int n_ChieuDai;`

Thành phần trước dấu “_” chỉ kiểu dữ liệu của biến (ví dụ: n = number), thành phần sau dấu “_” là tên biến (ví dụ: ChieuDai).

b) Đóng mở ngoặc {} trong 1 khối lệnh/ điều kiện/ hàm

Các dấu đóng, mở ngoặc {} đặt trên 1 dòng riêng biệt; dấu đóng ngoặc } phải thẳng hàng với dấu mở ngoặc {.

Các dòng lệnh nằm trong cặp đóng mở ngoặc {} phải được thụt vào 1 đoạn tab.

Ví dụ:

```
int main() {  
    cout<<"Truong Dai hoc Cong nghe Sai Gon."  
    return 0;  
}
```

3. Chương trình mẫu (0.4 điểm)

Tài nguyên: D:\NMLT\Lab1\Examples

a) Cho đoạn mã chương trình mẫu trong tập tin **Lab1a.cpp**. Nội dung của đoạn mã lệnh như sau:

STT	Mã lệnh
1.	//khai bao thu vien de su dung doi tuong cout
2.	#include "iostream"
3.	using namespace std;
4.	//chuong trinh chinh
5.	int main() { //bat dau khoi lenh cua chuong trinh chinh
6.	cout<<"Truong Dai hoc Cong nghe Sai Gon.";
7.	return 0;
8.	} //ket thuc chuong trinh chinh
9.	

Yêu cầu sinh viên:

Mở tập tin Lab1a.cpp trong thư mục trên.

- a.1/ Nhấn **Ctrl + F7** để kiểm lỗi chương trình; **F7** để dịch thành file **.exe**.
- a.2/ Nhấn **Ctrl + F5** để thực thi chương trình.
- a.3/ Ghi lại kết quả hiển thị trên màn hình vào giấy.
- a.4/ Bỏ dòng lệnh số **2** và biên dịch lại chương trình. Ghi nhận kết quả biên dịch vào giấy.

Bài 1: LÀM QUEN VỚI MÔI TRƯỜNG VISUAL C++ 2017

a.5/ Bỏ dòng lệnh số **3** và biên dịch lại chương trình. Ghi nhận kết quả biên dịch vào giấy.

b) Cho đoạn mã lệnh chương trình trong tập tin **Lab1b.cpp**. Nội dung của đoạn mã lệnh như sau:

STT	Mã lệnh
1.	<code>//Lab1b.cpp</code>
2.	<code>//khai bao thu vien de su dung doi tuong cout, cin</code>
3.	<code>#include <iostream></code>
4.	<code>using namespace std;</code>
5.	<code>(int main()) { //bat dau ham main</code>
6.	<code> //khai bao bien tensv co toi da 9 ki tu</code>
7.	<code> char tenSV[10];</code>
8.	<code> cout<<"Ban ten gi?";</code>
9.	<code> cin>>tenSV; //khong nhap duoc chuoi co khoang trang</code>
10.	
11.	<code> cout <<"\nTruong Dai hoc Cong Nghe Sai Gon"<<endl;</code>
12.	<code> cout <<"Khoa Cong Nghe Thong Tin"<<endl;</code>
13.	<code> cout <<"Chao ban :"<<tenSV<<endl;</code>
14.	<code> //xuat gia tri cua bien tensv ra man hinh</code>
15.	<code> return 0;</code>
16.	<code>//vi ham main duoc khai bao la int main nen phai co</code>
17.	<code>return</code>
18.	<code>} //ket thuc ham main</code>

Yêu cầu sinh viên:

b.1/ Thực thi từng dòng lệnh chương trình (F10). Viết lại **yêu cầu** của bài toán.

b.2/ Bỏ dòng lệnh `char ten[10]`, chương trình chạy được hay không? Vì sao?

b.3/ Bỏ dòng lệnh `return 0`, chương trình còn thực hiện được hay không? Vì sao?

b.4/ Tại dòng lệnh số **11**, hãy xóa bỏ chuỗi ký tự "**<<endl**" ra khỏi dòng lệnh. Biên dịch và chạy lại chương trình. Kết quả xuất ra như thế nào? Hãy ghi lại kết quả ra giấy. Giải thích ý nghĩa của từ "**endl**"

b.5/ Thay thế dòng lệnh **10** thành dòng lệnh: `cin.getline(tensv,10);` biên dịch và thực thi lại chương trình, nhập nội dung "**Ngoc Ha**"; ghi nhận lại kết quả của dòng lệnh **13**.

c) Cho đoạn mã lệnh được cài đặt trong tập tin **Lab1c.cpp**. Yêu cầu của đoạn mã lệnh là khi thực thi chương trình cho phép nhập vào **so_keo** và **so_bich**; sau đó tính số kẹo có trong số bịch đã nhập vào và xuất ra màn hình tổng số kẹo. Nội dung đoạn mã lệnh như sau:

STT	Mã lệnh
1.	<code>//Lab1c.cpp</code>
2.	<code>#include <iostream></code>
3.	<code>using namespace std;</code>
4.	<code>int main() { //bat dau ham main</code>
5.	<code> //khai bao 3 bien so nguyen so_keo, so_bich,</code>
6.	<code>tong_keo</code>
7.	<code> int so_Keo, so_Bich, tong_Keo;</code>
8.	<code> cout<<"Nhap so keo trong mot goi:";</code>
9.	<code> cin>>so_Keo;</code>
10.	<code> cout<<"\nNhap so bich keo dang co: ";</code>
11.	<code> cin>>so_Bich</code>
12.	<code> tong_Keo=so_Keo*so_Bich;</code>
13.	<code> cout<<"\nSo keo co trong <<so_Bich<<" bich</code>
14.	<code> la:"<<tong_Keo;</code>
15.	<code> return 0;</code>
	<code>} //ket thuc ham main</code>

Yêu cầu sinh viên:

c.1/ Biên dịch chương trình. Nhận xét? Nếu chương trình bị lỗi, hãy chỉ ra dòng lệnh bị lỗi là dòng nào và ghi nhận lại lỗi thông báo? Giải thích vì sao lỗi và hãy sửa lại cho đúng.

c.2/ Thực thi debug chương trình tại dòng lệnh chỉ định là dòng 11. Nhận xét?

c.3/ Viết chú thích cho các dòng lệnh 12, 13.

c.4/ Mở thư mục **Debug** và thực thi tập tin **Lab1.exe**; có xem được kết quả chương trình hay không? Về lại **Lab1c.cpp**, thêm vào dòng lệnh thứ **14** có nội dung như sau: **system("pause")**; biên dịch lại chương trình; chạy lại tập tin **Lab1.exe** có trong thư mục **Debug**; có xem được kết quả cuối cùng của chương trình hay không?

c.5/ Nếu bỏ dòng lệnh **12**, có thể sửa lại dòng lệnh **13** theo sau như thế nào để chương trình vẫn xuất ra được kết quả theo yêu cầu? Hãy viết dòng lệnh đó ra giấy.

4. Chỉnh sửa số liệu theo yêu cầu (0.2điểm)

a) Mở **Lab1b.cpp**, thêm vào một khai báo biến *So_thich* là chuỗi có tối đa 15 kí tự, cho phép nhập vào *Tên*, *Sở thích* của một người. Sau đó xuất ra màn hình nội dung như sau:

```
*****  
Ho ten: <Tên>  
So thich: <Sở thích>  
*****
```

Ví dụ: nhập tên là **Nam Anh**; sở thích là **Phim**; nội dung xuất ra màn hình như sau:

```
*****  
Ho ten: Nam Anh  
So thich: Phim  
*****
```

b) Mở *Lab1c.cpp*, thay đổi các dòng lệnh tương ứng cho phép nhập vào chiều dài, chiều rộng của một hình chữ nhật. Xuất ra màn hình chu vi và diện tích của hình chữ nhật đó. Nội dung xuất ra như sau:

```
CHUONG TRINH TINH CHU VI VA DIEN TICH HINH CHU NHAT
Chieu dai HCN : ...met
Chieu rong HCN: ... met
Dien tich HCN: ... met vuong
Chu vi HCN: ... met
```

5. Bài tập vận dụng (0.2 điểm)

Viết chương trình nhập vào 2 số nguyên, thực hiện các phép tính +, -, * của hai số trên. Giả sử, nếu nhập vào số thứ nhất là **6**, số thứ hai là **4** thì trên màn hình sẽ hiển thị kết quả sau:

```
Tổng của 2 số a = 6 và b = 4 là 10
Hiệu của 2 số a = 6 và b = 4 là 2
Tích của 2 số a = 6 và b = 4 là 24
```

IV. BÀI TẬP LÀM THÊM:

1. Viết chương trình nhập xuất ra màn hình thông tin của mỗi sinh viên theo mẫu sau:

```
Ho ten:
Ma so sinh vien:
Lop:
So dien thoai:                               Gioi tinh:
```

2. Viết chương trình nhập vào 2 số nguyên dương a và b, cho biết kết quả chia lấy phần nguyên và phần dư của a với b.

- Viết chương trình nhập một số nguyên dương N có 2 chữ số từ bàn phím, xuất ra màn hình tổng các chữ số của N. **Ví dụ:** Nhập N = 48, kết quả in ra màn hình là: $4+8=12$
- Viết chương trình cho phép nhập vào giờ, phút và giây. Hãy đổi sang giây và in kết quả ra màn hình.
- Viết chương trình nhập vào 4 số nguyên a, b, c, d. Tính giá trị trung bình cộng của 4 số trên và in kết quả ra màn hình.

BÀI 2: CÁC CẤU TRÚC ĐIỀU KHIỂN**I. MỤC TIÊU**

Sau khi thực hành xong bài này, sinh viên phải biết cách:

- Sử dụng lệnh **if** theo một hoặc nhiều điều kiện.
- Sử dụng lệnh **switch case** để viết **menu** cho chương trình.
- Giải các bài toán bằng máy tính với các điều kiện chọn và các quyết định khác nhau dựa trên các điều kiện khác nhau.

II. TÓM TẮT LÝ THUYẾT

- Qui ước:
 - `Expression, expression1, expression2, ...`: các biểu thức *điều kiện*
 - `Statement, statement1, statement2,...`: lệnh hoặc khối lệnh, nếu là khối lệnh thì được đặt trong cặp dấu `{ }`

1. Cấu trúc điều khiển if

Nội dung	Ví dụ
<ul style="list-style-type: none">▪ Lựa chọn phát biểu theo <u>một điều kiện</u> và chỉ có <u>một khả năng</u> xảy ra. <pre>if (expression) statement;</pre>	<pre>//nếu điểm >=5 thì xuất ra màn hình nội dung <i>Dau</i> if (diem>=5) cout<<"Dau";</pre>
<ul style="list-style-type: none">▪ Lựa chọn phát biểu theo <u>một điều kiện</u> và có <u>hai khả năng</u> xảy ra.	<pre>//nếu điểm >=5 thì xuất ra màn hình nội dung <i>Dau</i></pre>

<pre> if (expression) statement1; else statement2; </pre>	<p>Ngược lại xuất ra <i>Rot</i></p> <pre> if (diem>=5) cout<<"Dau"; else cout<<"Rot"; </pre>
<p>▪ Lựa chọn phát biểu theo <u>nhiều điều kiện</u></p> <pre> if (expression1) statement1; else if (expression2) statement2; else if (expressionn) statementn; else statement; </pre>	<pre> if (diem>=7) cout<<"Kha"; else if (diem>=5) cout<<"Trung binh"; else cout<<"Yeu"; </pre> <p>//Ý nghĩa của đoạn mã lệnh trên: nếu điểm>=7 xuất ra màn hình nội dung là "Khá". Ngược lại, nếu điểm >=5 xuất nội dung là "Trung bình". Ngược lại xuất nội dung là "Yếu"</p>

Lưu ý:

- **expression** phải để trong cặp ngoặc đơn "()";
- sau **expression** không có dấu chấm phẩy ";"

2. Cấu trúc *switch case*

Nội dung	Ví dụ
<pre>switch (expression){//integer case exp1 : statement1; //có thể 1 hoặc nhiều phát biểu case exp2 : statement2; ... case expn : statementn; default: statementn+1; }</pre>	<pre>int so; cout<<"\nNhap 1so:"; cin>>so; switch (so){ case 1: cout<<"So mot"; break; case 2: cout<<"so hai!"; break; case 3: cout<<"So ba"; break; default: cout<<"Ban nhap sai roi !"; break; }</pre>

Lưu ý:

- **expression** phải trả về giá trị nguyên.
- **expression** rơi vào **case** nào, sẽ thực hiện các phát biểu sau **case** đó đến khi nào gặp phát biểu **break** thì dừng.

III. NỘI DUNG THỰC HÀNH

Tài nguyên: D:\NMLT\Lab2\Examples

1. Chương trình mẫu (0.4 điểm)

a) Đây là đoạn mã lệnh được lưu trong tập tin *Lab2a.cpp*. Yêu cầu của đoạn chương trình là cho phép nhập vào **hai số nguyên a, b** và tìm **số lớn nhất** trong hai số đó.

STT	Mã lệnh
1.	<code>//Lab2a.cpp</code>
2.	<code>#include <iostream></code>
3.	<code>using namespace std;</code>
4.	<code>void main() {</code>
5.	<code> int a,b, max;</code>
6.	<code> cout<<"\n Nhập so thu nhat:";</code>
7.	<code> cin>>a;</code>
8.	<code> cout<<"\n Nhập so thu hai:";</code>
9.	<code> cin>>b;</code>
10.	<code> /*xet a co lon hon b hay khong</code>
11.	<code> neu a> b thi gan a cho bien max</code>
12.	<code> nguc lai gan b cho bien max</code>
13.	<code> */</code>
14.	<code> if (a>b)</code>
15.	<code> max=a;</code>
16.	<code> else</code>
17.	<code> max =b;</code>
18.	<code> cout<<"\n So lon nhat la: "<<max<<endl;</code>
19.	<code> system("pause");</code>
20.	<code>}</code>
21.	

Yêu cầu sinh viên:

Bài 2: CÁC CẤU TRÚC ĐIỀU KHIỂN

a.1/ Thực thi **từng bước** chương trình và viết ra giấy ý nghĩa của chương trình.

a.2/ Viết lại kết quả khi thực thi chương trình.

b) Đây là đoạn mã lệnh được lưu trong tập tin **Lab2b.cpp**. Yêu cầu của đoạn mã lệnh là cho phép nhập một số nguyên vào biến **ma_truong** là mã của một trường đại học; sau đó cho biết mã nhập vào là mã của trường có tên gì và địa chỉ của trường. Biết rằng, tên trường tương ứng với giá trị của **ma_truong** là:

1. Đại học Tài chính.
2. Đại học Kinh tế.
3. Đại học Khoa học Tự nhiên.
4. Đại học Bách Khoa.
5. Đại học Công Nghệ Sài Gòn.

STT	Mã lệnh
1.	<code>//lab2b</code>
2.	<code>#include <iostream></code>
3.	<code>using namespace std;</code>
4.	<code>void main() {</code>
5.	<code> int ma_Truong;</code>
6.	<code> cout<<"\nCho biet ma truong(1..5) :";</code>
7.	<code> cin>>ma_Truong;</code>
8.	<code> //kiem tra ma_truong co bang 1 hay khong</code>
9.	<code> if (ma_Truong==1) {</code>
10.	<code> cout<<"\nDai hoc Tai chinh.";</code>
11.	<code> cout<<"\nDia chi: 297 Nguyen Tri Phuong Quan 10";</code>
12.	<code> }</code>
13.	<code> //kiem tra ma_truong voi 2</code>
14.	<code> else if (ma_Truong==2) {</code>
15.	<code> cout<<"\nDai hoc Kinh Te.";</code>
16.	<code> cout<<"\nDia chi: 59C Nguyen Dinh Chieu Quan 3";</code>

```

17.     }
18.     //kiem tra ma_truong voi 3
19.     else if (ma_Truong==3) {
20.         cout<<"\nDai hoc Khoa hoc Tu nhien.";
21.         cout<<"\nDia chi: 227 Nguyen Van Cu Quan 5";
22.     }
23.     //kiem tra ma_truong voi 4
24.     else if (ma_Truong==4) {
25.         cout<<"\nDai hoc Bach khoa.";
26.         cout<<"\nDia chi: 268 Ly Thuong Kiet Quan 10";
27.     }
28.     //kiem tra ma_truong voi 5
29.     else if (ma_Truong==5) {
30.         cout<<"\nDai hoc Cong Nghe Sai Gon.";
31.         cout<<"\nDia chi: 180 Cao Lo Quan 8";
32.     }
33.     //ma_truong khong bang voi cac truong hop xet o tren
34.     else
35.         cout<<"\nGo sai ma truong!!!";
36.     system("pause");
37. }
38.

```

Yêu cầu sinh viên:

b.1/ Mở và thực thi từng dòng lệnh trong **lab2b.cpp**.

b.2/ Thực thi chương trình, nhập vào giá trị 3. Viết kết quả ra giấy.

b.3/ Thực thi lại chương trình và nhập vào giá trị 5. Viết kết quả ra giấy.

b.4/ Thực thi lại chương trình và nhập vào giá trị 9. Viết kết quả ra giấy.

c) Đây là đoạn mã lệnh được lưu trữ trong tập tin **Lab2c.cpp**. Yêu cầu của đoạn mã lệnh là nhập vào một số nguyên và xuất ra màn hình nội dung tương ứng khi nhập vào các giá trị số.

STT

Mã lệnh

```
1. //Lab2c
2. #include <iostream>
3. using namespace std;
4. void main() {
5.     int so;
6.     cout<<"Nhập vào 1 số nguyên:";
7.     cin>>so;
8.     switch(so) {
9.         case 0:
10.            cout<<"\nSố không";
11.            break;
12.         case 1:
13.         case 2:
14.            cout<<"\n Số 1 hoặc số 2";
15.            break;
16.         case 3:
17.            cout<<"\n Số 3 đây mà";
18.            break;
19.         case 4:
20.            cout<<"\n Số 4 đây nè";
21.            break;
22.         case 5:
23.         case 6:
24.            cout<<"\n Số 5 hoặc Số 6";
25.         default:
26.            cout<<"\n Nhưng số khác";
27.            break;
28.     }
29. }
30.
```

Yêu cầu sinh viên:

c.1/ Hãy viết lại kết quả của chương trình sau mỗi lần thực thi trong các trường hợp sau:

- Nhập 0
- Nhập 1
- Nhập 3
- Nhập 5
- Nhập 9.

d) Đây là đoạn mã lệnh được lưu trong tập tin **lab2d.cpp**. Yêu cầu của đoạn mã lệnh là khi thực thi chương trình, cho phép nhập vào Tên, điểm toán, điểm văn; sau đó tính điểm trung bình theo công thức: $dtb = (\text{điểm toán} * 2 + \text{điểm văn}) / 3$. Cho biết kết quả là đậu hay rớt, biết rằng nếu điểm trung bình ≥ 5 là đậu ngược lại là rớt; nếu là đậu, cho biết kết quả xếp loại, biết rằng:

- Nếu $dtb \geq 9$, xếp loại Xuất sắc
- Nếu $8 \leq dtb < 9$, xếp loại giỏi
- Nếu $7 \leq dtb < 8$, xếp loại khá
- Nếu $5 \leq dtb < 7$, xếp loại tb.

Nội dung đoạn mã lệnh như sau:

STT	Mã lệnh
1.	<code>//lab2d.cpp</code>
2.	<code>#include <iostream></code>
3.	<code>using namespace std;</code>
4.	<code>void main() {</code>
5.	<code> char ten[15];</code>
6.	<code> float diemToan, diemVan, diemTB;</code>
7.	<code> cout<<"\n Cho biet Ten hv:";</code>
8.	<code> cin.getline(ten,15);</code>
9.	<code> cout<<"\n Nhap Diem Toan:";</code>
10.	<code> cin>>diemToan;</code>
11.	<code> cout<<"\n Nhap Diem Van:";</code>
12.	<code> cin>>diemVan;</code>
13.	<code> diemTB=(diemToan*2+ diemVan)/3;//tinh diem trung</code>
14.	<code> binh</code>
15.	<code> //xu ly ket qua xep loai tai day.</code>
16.	<code> cout<<"\n Chuong trinh tinh diem trung binh"<<endl;</code>
17.	<code> cout<<"Ho ten: "<<ten<<endl;</code>
18.	<code> cout<<"Toan: "<<diemToan<<endl;</code>
19.	<code> cout<<"Van: "<<diemVan<<endl;</code>
20.	<code> cout<<"Diem TB: " <<diemTB<<endl;</code>
21.	<code> if(diemTB<5)</code>
22.	<code> cout<<"\nKet qua: Rot!"<<endl;</code>
23.	<code> else { //diem tb>=5</code>
24.	<code> cout<<"\nKet qua: Dau!"<<endl;</code>
25.	<code> if(diemTB>=9)</code>
26.	<code> cout<<"\nXep loai: Xuat sac"<<endl;</code>
27.	<code> else if(diemTB>=8)</code>
28.	<code> cout<<"\nXep loai: Gioi"<<endl;</code>
29.	<code> else if(diemTB>=7)</code>
30.	<code> cout<<"\nXep loai: Kha"<<endl;</code>
31.	<code> else</code>
32.	<code> cout<<"\nXep loai: Trung binh"<<endl;</code>

Bài 2: CÁC CẤU TRÚC ĐIỀU KHIỂN

33.	}
34.	}
35.	

Yêu cầu sinh viên:

d.1/ Biên dịch chương trình và thực thi. Viết lại các kết quả xuất ra màn hình khi nhập vào các trường hợp sau:

- Bui 5 7
- Nguyen 4 3
- Ly 8 9

d.2/ Tại dòng lệnh 28, có thể thay thế bởi dòng lệnh sau không? Giải thích?

```
else if (8<=diemTB<9)
```

d.3/ Tại các dòng lệnh 28, 30 có thể bỏ các từ khóa `else` được không? Giải thích.

2. Chỉnh sửa số liệu theo yêu cầu (0.2 điểm)

a) Mở `Lab2_1.cpp`, thực thi chương trình và nhập vào hai số 2 và 4. Sau đó thực thi lại chương trình và nhập lại hai số 0 và 4. Viết lại kết quả chương trình.

Dòng lệnh thứ 13 chỉ thi hành được khi a khác 0. Vì thế khi nhập a = 0, chương trình sẽ lỗi.

- Hãy viết lại nội dung báo lỗi khi thực thi chương trình.
- Hãy thêm điều kiện để chương trình vẫn chạy được với a=0.
- Hãy sửa lại chương trình để giải được phương trình $ax + b=0$.

b) Mở tập tin `lab2_2.cpp` và sửa lại đoạn mã lệnh để thực hiện được yêu cầu bài toán sau: nhập vào 2 số nguyên a, b; xây dựng menu chương trình với các chức năng:

1. Tính tổng a+b

2. Tính hiệu a-b
3. Tính tích a*b
4. Tính thương a/b

3. Bài tập vận dụng (0.2 điểm)

a) *Viết chương trình cho phép nhập vào một số nguyên. Kiểm tra xem số vừa nhập vào có là vừa là bội số của 3 vừa là bội số của 5 hay không?*

b) *Giả sử yêu cầu của bài toán trong **lab2d.cpp** thay đổi, hãy viết lại các dòng lệnh phù hợp để giải quyết yêu cầu bài toán theo tiêu chí như sau:*

- Nếu điểm trung bình ≥ 9 thì được xếp loại **Xuất sắc**. Ngược lại:
- Nếu điểm trung bình ≥ 8 và < 9 thì được xếp loại **Giỏi**. Ngược lại:
- Nếu điểm trung bình ≥ 6.5 và < 8 thì được xếp loại **Khá**. Ngược lại:
- Nếu điểm trung bình ≥ 5 và < 6.5 thì **Trung Bình**. Ngược lại:
- Nếu điểm trung bình < 5 thì được xếp loại **Yếu**.

Kết quả xuất ra màn hình như sau:

```
*****  
Ho ten: <ten>  
Diem Toan: <Diemtoan>  
Diem Van: <DiemVan>  
Diem Trung binh: <DiemTB>  
Xep Loi: <XL>  
*****
```

4. Bài tập tổng hợp (0.2 điểm)

Viết chương trình trên giấy (không sử dụng máy tính) các bài toán sau:

Bài 2: CÁC CẤU TRÚC ĐIỀU KHIỂN

Viết chương trình tính tiền cước TAXI. Biết rằng:

- *km đầu tiên là 15000^d.*
- *Mỗi km tiếp theo là 16500^d.*
- *Nếu lớn hơn 30km thì mỗi km thêm sẽ là 16000^d.*

Hãy nhập số km sau đó in ra số tiền phải trả.

Ví dụ: Giả sử số km đã đi là 37km thì số tiền phải trả là: $15000 + 29 * 16500 + (37 - 30) * 16000$

IV. BÀI TẬP LÀM THÊM

1. Nhập vào 3 số nguyên dương. Kiểm tra xem 3 số đó có lập thành tam giác không? Nếu có hãy cho biết tam giác đó thuộc loại nào? (Cân, vuông, vuông cân, đều, thường).
2. Viết chương trình nhập vào số nguyên dương n. Kiểm tra xem n có phải là số chính phương hay không? (số chính phương là số khi lấy căn bậc 2 có kết quả là nguyên).

Ví dụ: $n=4 \rightarrow n$ là số chính phương, $n=5 \rightarrow n$ không là số chính phương.

BÀI 3: CÁC CẤU TRÚC LẶP

I. MỤC TIÊU

Sau khi thực hành xong bài này, sinh viên phải biết cách:

- Sử dụng toán tử tăng (++) và toán giảm số (--)
- Sử dụng lệnh **for** để giải bài toán biết trước số lần lặp
- Sử dụng lệnh **while** để giải bài toán (có thể) chưa biết trước số lần lặp.

II. TÓM TẮT LÝ THUYẾT

Nội dung	Ví dụ
<p>Toán tử ++ : cộng thêm 1 vào giá trị đang có của biến.</p> <p>Toán tử-- : trừ đi 1 vào giá trị đang có của biến.</p> <p>Có 2 vị trí của toán tử ++ và -- :</p> <ul style="list-style-type: none">▪ prefix: giá trị của biến được cập nhật trước, thực hiện biểu thức sau.▪ postfix: thực hiện biểu thức trước, giá trị của biến được cập nhật sau. <p>Qui ước:</p> <ul style="list-style-type: none">▪ initialization: là biểu thức gán để tạo giá trị ban đầu cho biến điều khiển; còn được gọi là biểu thức khởi tạo.▪ test: là một biểu thức logic điều kiện để kiểm tra tiếp tục vòng lặp.▪ update: là biểu thức gán dùng để thay đổi giá trị của biến điều khiển vòng lặp.▪ expression: biểu thức điều kiện.	<pre>int x, y, z, t; x=4; y=7; z=++x + (--y/2); //x=5; y=6; z=8 t=z++ + x--; //t=13; z=9; x=4; //khởi tạo i=0 i=0; //kiểm tra i<n i<n; //cập nhật i i++; n>2</pre>

Bài 3: CÁC CẤU TRÚC LẶP

<p>a. Cấu trúc vòng lặp for:</p> <pre>for(initialization; test; update){ statement; } //Nếu chỉ có 1 phát biểu có thể bỏ cặp ngoặc // {} // }</pre> <p>Cơ chế làm việc:</p> <ul style="list-style-type: none">▪ Bước 1 : Thực hiện biểu thức initialization▪ Bước 2 : Đánh giá biểu thức test<ul style="list-style-type: none">○ Nếu true, thực hiện statement○ Nếu false, kết thúc vòng lặp▪ Bước 3 : Thực thi update, trở về bước 2 kiểm tra lại biểu thức test	<pre>int i; for (i=1;i<=10;i++) { cout<<"\nHoc tot!"; } //in ra 10 dòng //Hoc tot!</pre>
<p>Lưu ý:</p> <ul style="list-style-type: none">▪ Có thể bỏ qua biểu thức initialization nếu đã khởi động ở ngoài vòng for.▪ Có thể khai báo biến ngay trong initialization của for▪ Có thể bỏ qua biểu thức update nếu được thực hiện trong thân vòng lặp.▪ Bỏ qua biểu thức test có thể gây ra vòng lặp vô tận.	<pre>int i=0,s=0; for (;i<10;i++) s+=i; for(int i=0,s=0; i<10; i++) s+=i; for (int i=0, s=0;i<10;) s+=i++;</pre>

Bài 3: CÁC CẤU TRÚC LẶP

<ul style="list-style-type: none">▪ Phát biểu <code>break</code> dùng để thoát (kết thúc) vòng lặp hiện tại.	<pre>for (int i=0, s=0; i<10; i++) s+=i; int i=0, s=0; for (; i<10; i++) { s+=i; if (s>=20) break; }</pre>
<p>b. Cấu trúc lặp while:</p> <pre>while (expression){ statement; //nếu chỉ có 1 phát biểu có thể bỏ {} }</pre> <p>Cơ chế làm việc:</p> <p>Biểu thức điều kiện expression sẽ được xem xét :</p> <ul style="list-style-type: none">▪ Nếu có giá trị true thì phát biểu statement sẽ được thực hiện lại.▪ Nếu có giá trị false thì vòng lặp sẽ kết thúc công việc và chương trình sẽ thực hiện tiếp các lệnh sau phát biểu statement của vòng lặp while.	<pre>int i=1, s=0; while (i<10) { s+=i; i++; }</pre>
<p>Lưu ý :</p> <ul style="list-style-type: none">▪ Không có dấu chấm phẩy “;” sau expression▪ while là vòng lặp kiểm tra điều kiện trước	

(*pretest loop*) – *expression* sẽ được định giá trước khi vòng lặp thực thi.

- Trong đoạn chương trình thực thi của vòng lặp phải có tác vụ làm cho **expression** mang giá trị **false** để ngăn chặn trường hợp vòng lặp thực hiện không ngừng (*infinite loop*)
- Mỗi vòng lặp **for** đều có thể triển khai bằng vòng lặp **while**.

III. NỘI DUNG THỰC HÀNH

Tài nguyên: D:\NMLT\Lab3\Examples

1. Chương trình mẫu (0.4 điểm)

a) Đây là đoạn mã lệnh được lưu trong tập tin **Lab3_a.cpp**. Yêu cầu của đoạn mã lệnh là cho phép nhập một số nguyên n , kiểm tra điều kiện nếu $n \leq 0$ thì thoát khỏi chương trình, nếu $n > 0$ thì tính giá trị của biểu thức:

$$S = 1 + 2 + 3 + \dots + n$$

Nội dung của đoạn mã lệnh như sau:

STT	Mã lệnh
-----	---------

Bài 3: CÁC CẤU TRÚC LẬP

```
1. //Lab3_a.cpp
2. #include <iostream>
3. using namespace std;
4. void main() {
5.     int n,i,sum;
6.     cout<<"\nNhap so nguyen n:";
7.     cin>>n;
8.     if(n>0) {
9.         sum=0; //giai thich
10.        //duyet tu 1 den n
11.        for(i=1;i<=n;i++){
12.            sum+=i;//sum=sum+i
13.        }
14.        cout<<"\n Tong can tinh la: "<<sum<<endl;
15.    }
16.    else
17.        cout<<"\n So vua nhap la ko duong. Ket
18.    thuc!"<<endl;
19.    system("pause");
20. }
21.
22.
```

Yêu cầu sinh viên:

a.1/ Thực thi từng bước **Lab3_a.cpp** và ghi nhận giá trị vào bảng sau khi nhập n=6:

i	1	2	3	4	5	6
sum	?	?	?	?	?	?

a.2/ Nếu bỏ lệnh **sum=0** tại dòng lệnh số 11 chương trình có cho ra kết quả đúng không? Giải thích vì sao?

b) Đây là đoạn mã lệnh được lưu trong tập tin **Lab3_b.cpp**. Yêu cầu của đoạn mã lệnh là cho phép nhập vào một dãy số nguyên, chương trình sẽ dừng khi

Bài 3: CÁC CẤU TRÚC LẬP

nhập vào số 0. Sau đó, tính tổng các số nguyên vừa được nhập. Đây là nội dung của Lab3_b.cpp.

STT	Mã lệnh
1.	<code>//Lab3_b.cpp</code>
2.	<code>#include <iostream></code>
3.	<code>using namespace std;</code>
4.	<code>void main() {</code>
5.	<code> int n, sum=0;</code>
6.	<code> cout<<"\nNhap so nguyen n:";</code>
7.	<code> cin>>n;</code>
8.	<code> while (n!=0) {</code>
9.	<code> sum+=n; //sum=sum+n</code>
10.	<code> cout<<"\nNhap so nguyen khac 0.";</code>
11.	<code> cout<<"\nNhap 0 de thoat khoi chuong</code>
12.	<code> trinh.";</code>
13.	<code> cin>>n;</code>
14.	<code> }</code>
15.	<code> cout<<"\nTong cac so nguyen vua nhap la:</code>
16.	<code> "<<sum<<endl;</code>
17.	<code> system("pause");</code>
18.	<code>}</code>

Yêu cầu sinh viên:

b.1/ Mở và thực thi từng bước lab3_b.cpp và ghi nhận lại giá trị của biến **sum** sau khi mỗi lần nhập vào giá trị của **n** vào bảng sau:

n	?	?	?	?	?	?	?	?
sum	?	?	?	?	?	?	?	?

b.2/ Nếu bỏ dòng lệnh số **8**, chương trình còn thực thi được không? Giải thích?

c) Đây là đoạn mã lệnh được lưu trong tập tin **Lab3_c.cpp**. Yêu cầu của đoạn mã lệnh là cho phép nhập vào một số nguyên ≥ 2 . Kiểm tra xem số nguyên đó có

Bài 3: CÁC CẤU TRÚC LẬP

thỏa điều kiện là số hoàn thiện hay không? Biết rằng số hoàn thiện là số có tổng các ước số của nó (không kể nó) bằng chính số đó. Ví dụ: $n=6$; $1+2+3=6=n$
→ n là số hoàn thiện.

STT	Mã lệnh
1.	<code>//Lab3_c.cpp</code>
2.	<code>#include <iostream></code>
3.	<code>using namespace std;</code>
4.	<code>void main() {</code>
5.	<code> int n;</code>
6.	
7.	<code> cout<<"\nNhap so nguyen n>=2:";</code>
8.	<code> cin>>n;</code>
9.	<code> //kiem tra dieu kien de luon nhap duoc so nguyen>=2</code>
10.	<code> while (n<2) { //chu y bieu thuc dieu kien</code>
11.	<code> cout<<"\n Nhap so nguyen>=2: ";</code>
12.	<code> cin>>n;</code>
13.	<code> }</code>
14.	<code> //kiem tra xem so nguyen do la so hoan thien hay ko?</code>
15.	<code> int i, sum=0;</code>
16.	<code> //tinh tong cac uoc so cua n</code>
17.	<code> for (i=1; i<n; i++)</code>
18.	<code> if (n%i==0) //n chia het cho i</code>
19.	<code> sum+=i;</code>
20.	<code> if (sum==n)</code>
21.	<code> cout<<n<<" la so hoan thien!"<<endl;</code>
22.	<code> else</code>
23.	<code> cout<<n<<" khong la so hoan thien!"<<endl;</code>
24.	<code> system("pause");</code>
25.	<code>}</code>
26.	

Yêu cầu sinh viên:

c.1/ Thực thi **từng bước** tại dòng lệnh **while** (dòng lệnh số 10). Viết kết quả ra giấy khi nhập vào số 1, -5, 0, 9.

Bài 3: CÁC CẤU TRÚC LẶP

c.2/ Tại dòng lệnh 18, biểu thức điều kiện ở vòng lặp ($i < n$) có thể sửa thành $i \leq n$ được không? Giải thích?

c.3/ Dòng lệnh 19 có ý nghĩa gì?

c.4/ Thoát khỏi vòng lặp for, giá trị của **i** là bao nhiêu? Hãy viết kết quả ra giấy.

2. Chỉnh sửa số liệu theo yêu cầu (0.2 điểm)

a) Mở **Lab3_a.cpp**, hãy sửa lại chương trình để tính giá trị của biểu thức:

$$S = 1.2 + 2.3 + 3.4 + \dots + n.(n+1)$$

b) Mở **Lab3_b.cpp**, hãy sửa lại chương trình để chương trình cho phép nhập vào một dãy số nguyên, chương trình sẽ dừng khi nhập vào số **-999**. Thực hiện các yêu cầu sau:

- In ra các số nguyên chia hết cho 3 và 7.
- Tính tổng các số nguyên chia hết cho 3 và 7.

Hướng dẫn: sử dụng một biến tạm để lưu giá trị của tổng.

3. Bài tập vận dụng (0.2 điểm)

Viết chương trình cho phép nhập vào một số nguyên dương n . Tính giá trị của $S(n)$ với $n > 0$.
thức:

$$S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

Bài tập tổng hợp (0.2 điểm)

Viết chương trình trên giấy (không sử dụng máy tính) các bài toán sau:

Viết chương trình cho phép nhập vào một số nguyên dương n . Tính giá trị của $S(n)$ với $n > 0$.
thức:

$$S(n) = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots + \frac{n}{n+1}$$

IV. BÀI TẬP LÀM THÊM

- Viết chương trình nhập vào 1 số nguyên dương n , liệt kê ra các số nguyên dương là bội số của 2 hoặc là bội số của 5 mà nhỏ hơn n .

Ví dụ: $n=11$; các số nguyên dương là bội số của 2 hoặc là bội số của 5 mà nhỏ hơn 11 là: 2, 4, 5, 6, 8, 10.

- Viết chương trình đếm số ước số của số nguyên dương N .

Ví dụ: $N=12$; số ước số của 12 là 6

- Viết chương trình nhập vào hai số nguyên dương a và b . Tìm ước số chung lớn nhất và bội số chung nhỏ nhất của a và b .

Hướng dẫn: để tìm ước số chung lớn nhất của hai số nguyên dương a và b , ta áp dụng giải thuật sau:

Trong khi $a \neq b$ thì

 Nếu $a > b$ thì

$$a = a - b$$

 Ngược lại thì

$$b = b - a$$

Ví dụ: $a = 15$ và $b = 21$; vận dụng giải thuật trên, ta được giá trị của a , b qua các lần lặp như sau:

a	15	15	9	3
b	21	6	6	3

- Viết chương trình cho phép nhập vào một số nguyên dương n . Tính giá trị của biểu thức:

Bài 3: CÁC CẤU TRÚC LẬP

$$S(n) = \frac{1}{2} - \frac{2}{3} + \frac{3}{4} - \frac{4}{5} + \dots + (-1)^{n+1} \frac{n}{n+1} \quad \text{Với } n > 0$$

Yêu cầu: không sử dụng hàm pow(x,y).

- Viết chương trình nhập vào một số nguyên dương n, kiểm tra n có phải là số nguyên tố hay không? Biết rằng số nguyên tố là số chỉ có 2 ước số là 1 và chính nó.

BÀI 4: CẤU TRÚC LẶP DO...WHILE**I. MỤC TIÊU**

Sau khi thực hành xong bài này, sinh viên phải biết cách:

- Sử dụng cấu trúc lặp **dowhile** để giải bài toán theo số lần lặp (có thể) chưa biết trước.
- Sử dụng kết hợp cấu trúc lặp và các cấu trúc đã học để giải quyết bài toán.

II. TÓM TẮT LÝ THUYẾT

Nội dung	Ví dụ
<ul style="list-style-type: none">▪ statement: lệnh hoặc khối lệnh▪ expression: biểu thức điều kiện <p>Cấu trúc:</p> <pre>do{ statement; //có thể có nhiều lệnh }while (expression) ;</pre> <p>Cơ chế làm việc: thực hiện các statement lặp đi lặp lại khi expression còn true và sẽ ngưng khi expression là false.</p> <p>Lưu ý:</p>	<p>//đoạn mã lệnh bắt buộc nhập vào một số nguyên n>0.</p> <pre>int n; do{ cout<<"\nNhap n>0"; cin>>n; }while (n<=0) ;</pre>

- Có dấu chấm phẩy “;” sau (expression)
- Vòng lặp sử dụng cấu trúc **do ... while** luôn thực hiện ít nhất **01** lần (khác với while) do đây là cấu trúc lặp kiểm tra điều kiện **sau** khi thực hiện phát biểu.
- Sự khác biệt giữa các cấu trúc lặp :
 - for : thường được dùng cho số lần lặp **xác định**.
 - while : thường được dùng cho số lần lặp **chưa xác định** và điều kiện kiểm tra **trước**.
 - do...while : thường được dùng cho số lần lặp **chưa xác định** và điều kiện kiểm tra **sau**.
- Mỗi cấu trúc lặp có **đặc trưng riêng**, tuy nhiên hoàn toàn có thể sử dụng **bất kỳ** cấu trúc lặp nào cho bài toán **có tính chất lặp**.

III. NỘI DUNG THỰC HÀNH**1. Chương trình mẫu (0.4 điểm)**

Tài nguyên: **D:\NMLT\Lab4\Examples**

a) Đây là đoạn mã lệnh được lưu trong tập tin **Lab4_a.cpp**. Mục đích của đoạn mã lệnh là chương trình cho phép nhập vào một số nguyên $n > 0$ và tính giá trị biểu thức:

$$S = 1 + 2 + 3 + \dots + n$$

Sử dụng cấu trúc do ... while để kiểm tra điều kiện nhập vào của n.

Yêu cầu sinh viên

STT	Mã lệnh
1.	<code>//Lab4_a.cpp</code>
2.	<code>#include <iostream></code>
3.	<code>using namespace std;</code>
4.	<code>void main() {</code>
5.	<code> int n, i, sum=0;</code>
6.	<code> //nhap so nguyen n thoa dieu kien n>0</code>
7.	
8.	<code> do{</code>
9.	<code> cout<<"\nNhap so nguyen n:";</code>
10.	<code> cin>>n;</code>
11.	<code> }while (n<=0); //chu y dieu kien n phai thoa >0</code>
12.	<code> for(i=1, i<=n; i++) //duyet tu 1 den n</code>
13.	<code> sum+=i;</code>
14.	<code> cout<<"\nTong can tinh la: "<<sum<<endl;</code>
15.	<code> system("pause");</code>
16.	<code>}</code>
17.	

Bài 4: CẤU TRÚC LẶP DO ... WHILE

- a.1/** Biên dịch chương trình. Nhận xét? Nếu chương trình bị lỗi hãy cho biết dòng lệnh bị lỗi là dòng lệnh số mấy? Sửa lại dòng lệnh bị lỗi. Viết dòng lệnh đã sửa ra giấy.
- a.2/** Biên dịch và chạy từng bước chương trình và lần lượt nhập vào **-5, 0, 12**. Hãy viết lại kết quả thực thi chương trình ra giấy.
- a.3/** Đặt **breakpoint** tại dòng lệnh 13, debug chương trình, nhập $n=15$. Hãy ghi nhận giá trị của **i** và **sum** vào bảng giá trị sau:

i	1	7	9	12
sum				

Bài 4: CẤU TRÚC LẶP DO ... WHILE

b) Đây là chương trình cho phép nhập vào ba số nguyên a, b, c . Xuất ra màn hình giá trị của biểu thức: $2a + 3b + 4(a+c)b$. Chương trình sẽ dừng khi nhập vào phím Y hoặc y .

STT	Mã lệnh
1.	<code>//Lab4_b.cpp</code>
2.	<code>#include <iostream></code>
3.	<code>using namespace std;</code>
4.	<code>void main()</code>
5.	<code>{</code>
6.	<code> int a,b,c,s=0;</code>
7.	<code> char thoat;</code>
8.	<code> do{</code>
9.	<code> cout<<"\nNhap a:";</code>
10.	<code> cin>>a;</code>
11.	<code> cout<<"\nNhap b:";</code>
12.	<code> cin>>b;</code>
13.	<code> cout<<"\nNhap c:";</code>
14.	<code> cin>>c;</code>
15.	<code> s=2*a+3*b+4*(a+c)*b;</code>
16.	<code> cout<<"\nGia tri bieu thuc can tinh</code>
17.	<code>la:"<<s<<endl;</code>
18.	<code> cout<<"Ban muon dung? Nhap Y or y de dung.</code>
19.	<code>Chon phim:";</code>
20.	<code> cin>>thoat;</code>
21.	<code> }while((thoat!='Y') && (thoat !='y'));</code>
	<code> system("pause");</code>
	<code>}</code>

a) Đây là đoạn mã lệnh chương trình được lưu trong tập tin **lab4_c.cpp**, yêu cầu chương trình là cho phép nhập vào một số nguyên dương n . Kiểm tra xem n phải là số nguyên tố không? Nội dung đoạn mã lệnh như sau:

STT	Mã lệnh
1.	<code>//Lab4_c.cpp</code>
2.	<code>/*Kiem tra so nguyen duong n co phai la so nguyen to?</code>
3.	<code>So nguyen to la so nguyen khac 1, chi chia het cho 1</code>
4.	<code>va chinh no*/</code>
5.	<code>#include "iostream"</code>
6.	<code>using namespace std;</code>
7.	<code>void main()</code>
8.	<code>{</code>
9.	<code> int n;</code>
10.	<code> do{</code>
11.	<code> cout<<"\nNhap n>0:";</code>
12.	<code> cin>>n;</code>
13.	<code> }while(n<=0);</code>
14.	<code> if (n==2)</code>
15.	<code> cout<<"\n"<<n<<" la so nguyen to.";</code>
16.	<code> else if (n>2){</code>
17.	<code> int i=1;</code>
18.	<code> do{</code>
19.	<code> i++;</code>
20.	<code> }while ((i<n) && (n%i!=0));</code>
21.	<code> if (i==n)</code>
22.	<code> cout<<"\n"<<n<<" la so nguyen to.";</code>
23.	<code> else</code>
24.	<code> cout<<"\n"<<n<<" khong la so nguyen</code>
25.	<code>to.";</code>
26.	<code> }</code>
27.	<code> else</code>
28.	<code> cout<<"\nSo nguyen nhap vao <2.";</code>
29.	<code> system("pause");</code>
	<code>}</code>

Yêu cầu sinh viên:

Bài 4: CẤU TRÚC LẶP DO ... WHILE

a.1/ Hãy viết ra 10 số nguyên tố đầu tiên (không sử dụng chương trình).

a.2/ Hãy giải thích ý nghĩa của biểu thức điều kiện: **while ((i<n) && (n%i!=0));**

a.3/ Thực thi chương trình và nhập vào giá trị **n=15**. Ghi nhận giá trị của **i** vào bảng sau:

i	?	?	?	?
---	---	---	---	---

a.4/ Thực thi chương trình và nhập vào giá trị **n=19**. Ghi nhận giá trị của **i** vào bảng sau:

i	?	?	?	?
---	---	---	---	---	-----	-----	----	----

Ghi chú: số nguyên tố là số nguyên khác 1; chỉ **chia hết** cho **1 và chính nó**. Vì vậy ta sẽ sử dụng điều kiện này để làm điều kiện dừng vòng **do ... while**. Vòng lặp sẽ dừng khi số đang xét chia hết cho một số. Tuy nhiên, số chỉ được gọi là nguyên tố nếu số nó chia hết bằng chính nó.

2. Chỉnh sửa số liệu theo yêu cầu (0.2 điểm)

a) Mở **Lab4_a.cpp**, hãy sửa lại chương trình để cho phép nhập vào số nguyên n thỏa điều kiện $n > 0$ và $n < 100$. Tính giá trị của biểu thức:

$$S = 1.2 + 2.3 + 3.4 + \dots + n.(n+1)$$

Yêu cầu: giải quyết bài toán bằng cách sử dụng cấu trúc lặp **do...while**.

b) Mở **Lab4_b.cpp**, hãy sửa lại chương trình để chương trình cho phép thực hiện lặp đi lặp lại công việc dưới đây. Chương trình sẽ dừng khi nhập vào phím **N** hoặc **n**.

- Nhập vào hai số nguyên **a, b**.
- Xét giá trị của **b**:
 - Nếu **b** khác 0 thì xuất ra màn hình giá trị **-a/b**.
 - Ngược lại: xuất ra màn hình nội dung “Không thực hiện được”.

3. Bài tập vận dụng (0.2 điểm)

Viết chương trình cho phép nhập vào một số nguyên dương n . Tính giá trị

của biểu thức:
$$S = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots + \frac{n}{n+1}$$

Yêu cầu: giải bài toán trong **ba** trường hợp: sử dụng vòng lặp **do ... while**; **while** và **for**.

4. Bài tập tổng hợp (0.2 điểm)

Viết chương trình trên giấy (không sử dụng máy tính) các bài toán sau:

Viết chương trình cho phép nhập vào một số nguyên dương n thỏa điều kiện ≥ 2 và ≤ 100 . In ra tất cả các số nguyên tố nhỏ hơn n .

Hướng dẫn: sử dụng **lab4_c.cpp** để kiểm tra một số có là nguyên tố hay không.

IV. BÀI TẬP LÀM THÊM

1. Viết chương trình cho phép nhập vào 1 số nguyên $n > 0$. Cho biết số n là số có bao nhiêu chữ số. Ví dụ: $n = 1234$ là số có 4 chữ số.
2. Viết chương trình cho phép nhập vào 1 số nguyên $n > 0$. Tính tổng các chữ số có trong n . Ví dụ: $n = 1234$. Tổng các chữ số là: $1 + 2 + 3 + 4 = 10$.
3. Viết chương trình cho phép nhập vào 1 số nguyên $n > 0$. In ra màn hình n số nguyên tố đầu tiên.

BÀI 5: HÀM – CƠ CHẾ TRUYỀN THAM TRỊ**I. MỤC TIÊU**

Sau khi thực hành xong bài này, sinh viên phải biết cách:

- Xây dựng hàm có tham số được truyền theo cơ chế truyền tham trị.
- Tạo 3 file trong một project
- Gọi thực thi hàm.

II. TÓM TẮT LÝ THUYẾT

Nội dung	Ví dụ
<p>Hàm (<i>function</i>) : là một tập hợp các phát biểu (lệnh của chương trình) để thực hiện một công việc (có thể nhỏ hoặc lớn).</p> <p>Hàm được xây dựng <u>một lần</u> và được sử dụng <u>hiều lần</u>.</p>	

III. NỘI DUNG THỰC HÀNH**1. Chương trình mẫu (0.4 điểm)**

a) Đây là đoạn mã lệnh chương trình được lưu trong tập tin **Lab5_a.cpp**. Yêu cầu của đoạn chương trình là cho phép nhập vào 1 số nguyên dương k . In ra màn hình các số nguyên từ 1 đến k .

Ví dụ: Nhập $k=5$. In ra: 12345

Yêu cầu chương trình: xây dựng hàm `lietke_SoNguyen` để in ra màn hình dòng `12...k`.

<p>Khai báo hàm (function definition) : bao gồm các phát biểu (lệnh chương trình) để tạo nên hàm. Bao gồm các phần sau :</p> <ul style="list-style-type: none">▪ Kiểu dữ liệu trả về (return type). Ví dụ: int, char, float, ...▪ Tên hàm (function name). Ví dụ: nhập, kiểmtra, insoduong,...▪ Danh sách tham số (Parameter list). Có thể có hoặc không.▪ Thân hàm (function body): tập các lệnh	<pre>int kiểmtra(int n); // Hàm có tên là <i>kiểmtra</i>, có một tham số kiểu int có tên là n. Hàm có kiểu trả về là kiểu int.</pre>
<p>Gọi hàm (function call): Gọi thi hành hàm từ chương trình chính hay một hàm khác. Cú pháp: Function_name(Parameter list);</p> <p>Giá trị truyền vào hàm thông qua tham số hình thức có hai cách truyền theo tham biến và tham trị.</p> <p>Ở phạm vi bài thực hành này, sinh viên chỉ thực hành xây dựng hàm và gọi hàm theo cơ chế truyền tham trị. Khai báo hàm có tham số hình thức</p>	<pre>void kiểm_tra_nguyen_to(int n);</pre> <p>n ở đây là tham số hình thức và được truyền theo kiểu tham trị.</p>

<p>theo cơ chế truyền tham trị như sau:</p> <p><kiểu dữ liệu trả về> <Tên hàm> (<kiểu dữ liệu> <tên tham số>[;<...>])</p> <p>Hàm được sử dụng theo cơ chế truyền tham trị: giá trị của tham số thực không thay đổi so với trước khi gọi hàm.</p>	
<p>Lưu ý:</p> <ul style="list-style-type: none">▪ Hàm nên được chú thích để :<ul style="list-style-type: none">- Nêu rõ mục đích của hàm- Ý nghĩa các giá trị cần nhập vào hàm- Ý nghĩa các giá trị sẽ xuất ra của hàm▪ Nếu hàm có trả về một giá trị, thì kiểu dữ liệu của giá trị này phải được định nghĩa ở function header của hàm.▪ Nếu hàm không trả về giá trị nào hết thì có thể dùng void trên header.▪ Tên hàm thường được bắt đầu bằng một hành động, các từ tiếp	<p>int demSoNguyenTo(...) int dem_SoNguyenTo(...)</p>

- File thuvien_lab_5.h như sau

Bài 5: HÀM – CƠ CHẾ TRUYỀN THAM TRỊ

theo chỉ đối tượng mà hành động đó thực hiện

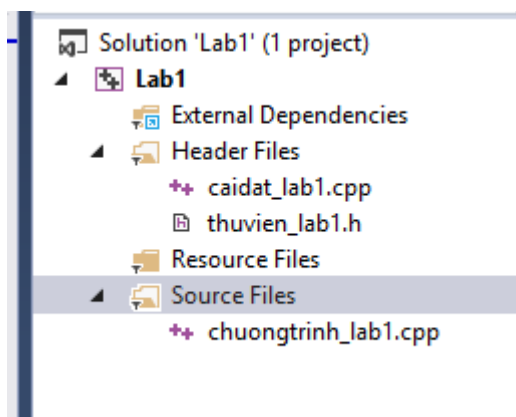
Trong môn học này, chúng ta sẽ thống nhất trong một project sẽ tiến hành cài đặt theo 3 file: gồm có: `thuvien_lab_n.h`, `caidat_lab_n.cpp`, `chuongtrinh_lab_n.cpp`.

Lưu ý: n ở đây là thứ tự các bài lab

- **`thuvien_lab_n.h`:** Chứa các thư viện cần khai báo của chương trình
- **`caidat_lab_n.cpp`:** Chứa các hàm cài đặt của chương trình
- **`chuongtrinh_lab_n.cpp`:** Chứa `void main()` để chạy chương trình

✚ Ví dụ về cách tạo 3 file trong một chương trình

- **Tạo một project ban đầu:**



Mã lệnh

```
#include<iostream>
using namespace std;
void lietKe_SoNguyen(int n);
```

- File `caidat_lab_5.cpp` như sau:

Mã lệnh

```
#include"thuvien_lab_5.h"
void lietKe_SoNguyen(int n){
    int i;
    cout << endl;
    for (i = 1; i <= n; i++)
        cout << i << " ";
}
```

- File chuongtrinh_lab_5.cpp như sau:

Mã lệnh

```
#include"thuvien_lab_5.h"

void main(){
    int k;
    do{
        cout << endl << "Nhap so nguyen duong: ";
        cin >> k;
    } while (k <= 0);
    lietKe_SoNguyen(k);
    system("pause");
}
```

Yêu cầu sinh viên:

- Nếu muốn in ra là **54321** thì sẽ viết lại dòng lệnh nào? Hãy sửa lại lệnh tương ứng để chương trình in ra được 54321.
- Đây là đoạn mã lệnh được lưu trong tập tin **Lab5_b.cpp**.

- Tập tin thư viện như sau:

Mã lệnh

```
#include<iostream>
using namespace std;
void capNhat_Bien(int a, int b);
```

Bài 5: HÀM – CƠ CHẾ TRUYỀN THAM TRỊ

- Tập tin cài đặt như sau:

```
Mã lệnh
#include"thuvien_lab_5.h"
void capNhat_Bien(int a, int b){
    if (a>b && a % 3 == 0)
        a += b - 3;
    if (b>5)
        b += b / 5;
    cout << "\nTrong ham cap nhat: a=" << a << " b="
<< b << endl;
}
```

- Tập tin chương trình như sau:

```
Mã lệnh
#include"thuvien_lab_5.h"

void main(){
    int n, m;
    cout << "\nNhap so nguyen thu nhat : ";
    cin >> n;
    cout << "\nNhap so nguyen thu hai : ";
    cin >> m;
    cout << "\nHai so nguyen nhap vao la: " << n <<
"\t" << m << endl;
    //goi ham capnhat_bien
    capNhat_Bien(n, m);
    cout << "\nHai so nguyen cap nhat moi: " << n <<
"\t" << m << endl;
    system("pause");
}
```

Yêu cầu sinh viên:

- b.1/** Ghi nhận kết quả của biến n, m trước và sau khi gọi hàm.
- b.2/** Giá trị của biến n, m sau khi gọi hàm có thay đổi không? Giải thích.

Bài 5: HÀM – CƠ CHẾ TRUYỀN THAM TRỊ

c) Đây là đoạn mã lệnh được lưu trong tập tin **Lab5_c.cpp**. Yêu cầu của đoạn chương trình là cho phép nhập vào một số nguyên dương **h** là chiều cao của tam giác. In ra màn hình tam giác số như sau (ứng với $h=5$)

1
12
123
1234
12345

Yêu cầu:

- Xây dựng hàm **in_TamGiacSo** để in ra tam giác như trên.
 - Sử dụng lại hàm **lietke_SoNguyen** để in tam giác trên.
- Tập tin thư viện như sau:

Mã lệnh
<pre>#include<iostream> using namespace std; void lietKe_SoNguyen(int n); void in_TamGiacSo(int height);</pre>

- Tập tin cài đặt như sau:

Mã lệnh
<pre>#include"thuvien_lab_5.h" void lietKe_SoNguyen(int n){ int i; cout << endl; for (i = 1; i <= n; i++) cout << i << " "; } void in_TamGiacSo(int height){ for (int i = 1; i <= height; i++) lietKe_SoNguyen(i); cout << endl; }</pre>

- Tập tin chương trình:

Mã lệnh
<pre>#include"thuvien_lab_5.h"</pre>

```

void main(){
    int h;
    do{
        cout << endl << "Nhap chieu cao tam giac so:
";
        cin >> h;
    } while (h <= 0);
    in_TamGiacSo(h);
    system("pause");
}

```

Yêu cầu sinh viên:

c.1/ Mở và thực thi **lab5_c.cpp**. Chương trình có gây lỗi hay không? Nếu có, hãy viết ra giấy dòng lệnh gây lỗi và sửa lại dòng lệnh gây lỗi đó, viết lại dòng lệnh đã sửa ra giấy.

```

12345
1234
123
12
.

```

c.2/ Nếu muốn in ra tam giác hình bên thì phải sửa lại dòng lệnh nào? ở hàm nào? Viết lại dòng lệnh đã sửa.

2. Chỉnh sửa số liệu theo yêu cầu (0.2 điểm)

a) Mở **Lab5_a.cpp**, hãy sửa lại hàm **lietke_SoNguyen** thành hàm **in_SoNguyen** cho phép in ra màn hình k lần số k . Ví dụ: $k=5$. In ra màn hình: 55555

Lưu ý: Không xây dựng thêm hàm mới.

b) Mở **Lab5_c.cpp**, hãy sửa lại hàm **lietke_SoNguyen** thành hàm **in_kitu** và hàm **in_TamGiacSo** thành hàm **in_TamGiacKiTu** để in được tam giác như sau: ví dụ, khi $h=5$

```

*
**
***
****
*****

```

Lưu ý: Không xây dựng thêm hàm mới.

3. Bài tập vận dụng (0.2 điểm)

Viết chương trình cho phép nhập vào chiều dài và chiều rộng của một hình chữ nhật. In ra màn hình hình chữ nhật trên theo dạng như sau:

Ví dụ: nhập chiều dài bằng 6, chiều rộng bằng 4.

```

*   *   *   *   *   *
*   *   *   *   *   *
*   *   *   *   *   *
*   *   *   *   *   *

```

Yêu cầu: xây dựng hàm **vehcn(int dai, int rong)** để vẽ được hình trên.

Gợi ý: sử dụng hàm **in_ki tu** ở câu **2b** để vẽ hình chữ nhật trên.

4. Bài tập tổng hợp (0.2 điểm)

Viết chương trình trên giấy (không sử dụng máy tính) các bài toán sau:

- Viết chương trình cho phép nhập vào một số nguyên dương h là chiều cao của tam giác. In ra màn hình tam giác như sau (ví dụ khi nhập $h=5$)

1
21
321
4321
54321

Yêu cầu:

- Xây dựng hàm **in_TamGiacSo_Giam** để in ra tam giác như trên.
 - Tương tự hàm **lietke_SoNguyen** ở **lab5_a.cpp**
- Viết chương trình cho phép nhập vào chiều dài và chiều rộng của một hình chữ nhật. In ra màn hình hình chữ nhật trên theo dạng như sau:

Ví dụ: nhập chiều dài bằng 6, chiều rộng bằng 4.

```

*   *   *   *   *   *
*
*
*
*   *   *   *   *   *

```

IV. BÀI TẬP LÀM THÊM

1. Viết chương trình tính diện tích và chu vi của hình chữ nhật với chiều dài và chiều rộng được nhập từ bàn phím.
2. Viết chương trình tính diện tích và chu vi hình tròn với bán kính được nhập từ bàn phím.
3. Nhập số nguyên dương n ($n > 0$). Liệt kê tất cả các số nguyên tố nhỏ hơn n .
4. Nhập số nguyên dương n ($n > 0$). Liệt kê n số chính phương đầu tiên.
5. Nhập số nguyên dương n ($n > 0$). Đếm xem có bao nhiêu số hoàn thiện $< n$.

BÀI 6: HÀM – CƠ CHẾ TRUYỀN THAM BIẾN**I. MỤC TIÊU**

Sau khi thực hành xong bài này, sinh viên phải biết cách:

- Xây dựng hàm có tham số được truyền theo hình thức tham biến.
- Gọi thực thi hàm.

II. TÓM TẮT LÝ THUYẾT

Nội dung	Ví dụ
<p>Khai báo hàm có tham số được truyền theo cơ chế tham biến:</p> <pre><kieu du lieu tra ve> <ten ham> (<kieu du lieu> &<ten tham so>[, <kieu du lieu> &<ten tham so 2>, ...])</pre>	<pre>void hoanvi (int &a, int &b);</pre>
<p>Gọi hàm (<i>function call</i>): Gọi thi hành hàm từ chương trình chính hay một hàm khác.</p> <p>Cú pháp:</p> <pre><ten ham>(danh sách tham số);</pre>	<pre>int a=5, b=9; hoanvi (a,b);</pre>

III. NỘI DUNG THỰC HÀNH**1. Chương trình mẫu (0.4 điểm)**

a) Đây là đoạn mã lệnh chương trình được lưu trong tập tin *lab6_a.cpp*, yêu cầu của chương trình là cho phép nhập vào 2 số nguyên

Bài 6: HÀM – CƠ CHẾ TRUYỀN THAM BIẾN

a, b và hoán đổi giá trị của 2 số *a* và *b* cho nhau. Xuất ra màn hình giá trị của *a, b* trước và sau khi hoán đổi.

Yêu cầu chương trình: xây dựng hàm **hoán vị** để hoán đổi giá trị *a, b*.

- File thư viện như sau:

Mã lệnh
<pre>#include<iostream> using namespace std; void hoanVi(int, int);</pre>

- File cài đặt như sau:

Mã lệnh
<pre>#include"thuvien_lab_6.h" void hoanVi(int n, int m){ int t = m; m = n; n = t; }</pre>

- File chương trình:

Mã lệnh
<pre>#include"thuvien_lab_6.h" void main(){ int a, b; cout << "\n Nhap so a:"; cin >> a; cout << "\n Nhap so b:"; cin >> b; cout << "\nGia tri truooc khi goi ham hoanvi la: "; cout << "a= " << a << " b= " << b << endl; hoanVi(a, b); //goi ham hoan vi 2 so a va b cout << "\nGia tri sau khi hoan doi a va b la: "; cout << "a= " << a << " b= " << b << endl;</pre>

```
system("pause");  
}
```

Yêu cầu sinh viên:

- a.1/ Chạy từng bước **lab6_a.cpp**.
- a.2/ Viết chú thích từng dòng lệnh trong hàm hoán vị.
- a.3/ Viết lại kết quả khi thực thi chương trình.
- a.4/ Nhận xét về giá trị của a, b trước và sau khi gọi hàm hoán vị.
- a.5/ Hàm hoán vị được xây dựng trên cơ sở giá trị của tham số được truyền theo cơ chế nào?

b) Đây là đoạn mã lệnh chương trình được lưu trong tập tin **lab6_b.cpp**, yêu cầu của chương trình là cho phép nhập vào một số nguyên $n > 0$ và tính giá trị biểu thức:

$$S = 1 + 2 + 3 + \dots + n$$

Yêu cầu chương trình:

- Xây dựng hàm **nhap** để nhập giá trị n.
- Xây dựng hàm **tinh_BieuThuc** để tính giá trị của biểu thức.
 - File thư viện như sau:

Mã lệnh

```
#include<iostream>  
using namespace std;  
//khai bao ham nhap  
void nhap_SoNguyen(int &);  
//khai bao ham tinh_BieuThuc  
long tinh_BieuThuc(int);
```

- File cài đặt như sau:

Mã lệnh

```
#include"thuvien_lab_6.h"
void nhap_SoNguyen(int &m){
    do{
        cout << "\nNhap so nguyen duong ...";
        cin >> m;
    } while (m <= 0);
}
long tinh_BieuThuc(int n){
    long s = 0;
    for (int i = 1; i <= n; i++)
        s += (long)i;
    return s;//goi tra ve gia tri cho ham
}
```

- File chương trình như sau:

Mã lệnh

```
#include"thuvien_lab_6.h"
void main(){
    int n;
    n = 0;
    //goi ham nhap so nguyen
    nhap_SoNguyen(n);
    //goi ham tinh bieu thuc va gan cho gia tri bien
    s
    long s = tinh_BieuThuc(n);
    cout << "\nGia tri bieu thuc la: " << s << endl;
    system("pause");
}
```

Yêu cầu sinh viên

b.1/ Mở và thực thi **lab6_b.cpp**.

b.2/ Tham số trong hàm **nhap_SoNguyen** được xây dựng theo cơ chế truyền tham trị hay tham biến? Giải thích vì sao?

Bài 6: HÀM – CƠ CHẾ TRUYỀN THAM BIẾN

b.3/ Thực thi chương trình và nhập $n=7$, ghi nhận giá trị của biến n vào bảng sau:

Trước khi gọi hàm nhap_SoNguyen	Trong thân hàm nhap_SoNguyen	Sau khi gọi hàm nhap_SoNguyen
$n=?$	$n=?$	$n=?$

b.4/ Nếu bỏ dòng khai báo (**long s=0**) trong hàm **tinh_BieuThuc** thì chương trình còn chạy được hay không? Vì sao?

b.5/ Nếu bỏ dòng lệnh (**return s**) trong thân hàm **tinh_BieuThuc** thì chương trình còn thực hiện được hay không? Vì sao?

c) Đây là đoạn mã lệnh chương trình được lưu trong tập tin *lab6_c.cpp*, yêu cầu của chương trình là cho phép nhập vào hai số nguyên a, b là hệ số của phương trình $ax+b=0$, cho biết nghiệm của phương trình.

Yêu cầu chương trình:

Xây dựng hàm **tim_NghiemPT_BacNhat** để tìm nghiệm của phương trình với quy ước:

- Hàm trả về -1, phương trình vô nghiệm
 - Hàm trả về 0, phương trình có vô số nghiệm
 - Hàm trả về 1, phương trình có 1 nghiệm $x=-b/a$
- File thư viện như sau:

Mã lệnh
<pre>#include<iostream> using namespace std; //khai bao ham nhap void nhap_SoNguyen(int &); int tim_Nghiem_PT BacNhat(int a, int b, float &t);</pre>

- File cài đặt như sau:

Mã lệnh
<pre>#include"thuvien_lab_6.h" void nhap_SoNguyen(int &m){</pre>

```

do{
    cout << "\nNhap so nguyen duong ...";
    cin >> m;
} while (m <= 0);
}
int tim_Nghiem_PT BacNhat(int a, int b, float &t){
    if (a == 0){
        if (b == 0)
            return 0;
        else
            return -1;
    }
    else{
        t = (float)-b / a;
        return 1;
    }
}

```

- File chương trình như sau:

Mã lệnh

```

#include"thuvien_lab_6.h"
void main(){
    int a, b;
    cout << "\nNhap gia tri a : " << endl;
    nhap_SoNguyen(a);
    cout << "\nNhap gia tri b : " << endl;
    nhap_SoNguyen(b);
    float x;
    int k = tim_Nghiem_PT BacNhat(a, b, x);
    if (k == -1)
        cout << "\nPhuong trinh vo nghiem!";
    else if (k == 0)
        cout << "\nPhuong trinh co vo so nghiem!";
    else//k=1
        cout << "\nPhuong trinh co nghiem duy nhat
x = " << x << endl;
    system("pause");
}

```

Yêu cầu sinh viên

c.1/ Mở và thực thi **lab6_c.cpp**.

c.2/ Tham số trong hàm **tim_Nghiem_PT BacNhat** được xây dựng theo cơ chế truyền tham trị hay tham biến? Giải thích vì sao?

c.3/ Có thể thay thế thành $t=-b/a$ được hay không? Giải thích?

c.4/ Có thể khai báo thành `int tim_Nghiem_PT BacNhat (int a, int b, float t)` hay không? Giải thích vì sao?

d) Đây là đoạn mã lệnh chương trình được lưu trong tập tin **lab6_d.cpp**, yêu cầu của chương trình là cho phép nhập vào một số nguyên dương n . Kiểm tra xem n phải là số nguyên tố không? Nếu n không phải là số nguyên tố, in ra các ước số của n và bé hơn n .

Yêu cầu chương trình:

- Xây dựng hàm **nhap_SoNguyenDuong** để nhập giá trị n . (tương tự **lab6_b.cpp**)
- Xây dựng hàm **kiemTra_SoNguyenTo** để kiểm tra xem n có phải là số nguyên tố không.
 - Nếu n là số nguyên tố thì hàm sẽ trả về giá trị là 1.
 - Nếu n không phải là số nguyên tố thì hàm sẽ trả về giá trị là 0.

Bài 6: HÀM – CƠ CHẾ TRUYỀN THAM BIẾN

- File thư viện như sau:

Mã lệnh

```
#include<iostream>
using namespace std;
//khai bao ham nhap
void nhap_SoNguyenDuong(int &);
int kiemTra_SoNguyenTo(int k);
void in_UocSo(int k);
```

- File cài đặt như sau:

Mã lệnh

```
#include"thuvien_lab_6.h"
void nhap_SoNguyenDuong(int &m){
    do{
        cout << "\nNhap so nguyen duong ...";
        cin >> m;
    } while (m <= 0);
}
int kiemTra_SoNguyenTo(int k){
    if (k<2)
        return 0;
    for (int i = 2; i<k; i++)
        if (k%i == 0)
            return 0;
    return 1;
}
/*In ra cac uoc so cua k <k*/
void in_UocSo(int k){
    cout << endl;
    for (int i = 1; i<k; i++)
        if (k%i == 0)
            cout << i << "\t";
    cout << endl;
```

```
}
```

- File chương trình như sau:

Mã lệnh
<pre>#include"thuvien_lab_6.h" void main(){ int n; //goi ham nhap so nguyen duong nhap_SoNguyenDuong(n); if (kiemTra_SoNguyenTo(n) == 1) cout << "\n" << n << " la so nguyen to." << endl; else{ cout << "\n" << n << " khong la so nguyen to."; cout << "\nCac uoc so < " << n << " la: "; in_UocSo(n); } system("pause"); }</pre>

Yêu cầu sinh viên

d.1/ Mở và thực thi **lab6_d.cpp**.

d.2/ Dòng lệnh 15 có thể viết lại theo cách khác không? Nếu có, hãy viết dòng lệnh đó ra giấy.

d.3/ Hàm `kiemtra_SoNguyenTo` và hàm `in_UocSo` là hai hàm có kiểu trả về giống nhau hay khác nhau? Cách gọi hai hàm này giống nhau hay khác nhau? Nếu khác nhau hãy chỉ rõ điểm khác nhau.

2. Chỉnh sửa số liệu theo yêu cầu (0.2 điểm)

a) Mở **Lab6_a.cpp** và sửa lại hàm hoán vị để giá trị của a và b sẽ được thay đổi sau khi gọi hàm. Viết lại hàm hoán vị ra giấy.

b) Mở **Lab6_b.cpp** và lưu lại với tên mới là **Lab6_b_1.cpp**, hãy sửa lại chương trình để cho phép nhập vào số nguyên n thỏa điều kiện $n > 0$ và $n < 1000$. Tính giá trị của biểu thức:

$$S = \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n+1}, \text{ với } n > 0$$

3. Bài tập vận dụng (0.2 điểm)

Viết chương trình cho phép nhập vào một số nguyên dương n . Tính giá trị của biểu thức :

$$S = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots + \frac{n}{n+1}, \text{ với } n > 0$$

Yêu cầu chương trình phải:

- Xây dựng hàm **nhap_SoNguyenDuong** để nhập giá trị n .
- Xây dựng hàm **tinh_BieuThuc** trả về giá trị của biểu thức.

4. Bài tập tổng hợp (0.2 điểm)

Viết chương trình trên giấy (không sử dụng máy tính) các bài toán sau:

Viết chương trình cho phép nhập vào một số nguyên dương n thỏa điều kiện $n \geq 2$ và $n \leq 100$. In ra tất cả các số nguyên tố nhỏ hơn n .

Ví dụ: nhập vào $n=12$. Sẽ in ra các số nguyên tố nhỏ hơn 12 là: 2 3 5 7 11

Gợi ý: sử dụng vòng lặp kiểm tra tất cả các số từ 2 đến $n-1$. Nếu số nào thỏa điều kiện là số nguyên tố thì xuất ra màn hình.

IV. BÀI TẬP LÀM THÊM

1. Viết chương trình tính diện tích và chu vi hình tròn với bán kính được nhập từ bàn phím.
2. Nhập số nguyên dương n ($n > 0$). Liệt kê n số chính phương đầu tiên. Biết rằng số chính phương là số có căn bậc hai là một số nguyên.

Ví dụ: $n=9$, n là số chính phương vì căn bậc hai của 9 là 3.

$n=10$ không là số chính phương, vì căn bậc hai của 10 không là số nguyên.

3. Nhập số nguyên dương n ($n > 0$). Đếm xem có bao nhiêu số hoàn thiện $< n$.
4. Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), đếm xem n có bao nhiêu chữ số chẵn và bao nhiêu chữ số lẻ.
5. Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), đếm xem n có bao nhiêu chữ số là số nguyên tố.

BÀI 7: MẢNG MỘT CHIỀU**I. MỤC TIÊU**

Sau khi thực hành xong bài này, sinh viên phải biết cách:

- Định nghĩa và thao tác được trên mảng một chiều.

II. TÓM TẮT LÝ THUYẾT

Nội dung	Ví dụ
<ul style="list-style-type: none">▪ Khai báo mảng một chiều: <code><kiểu dữ liệu> <tên mảng>[<số phần tử>;</code> Lưu ý:<ul style="list-style-type: none">- Các phần tử của mảng sẽ được tính từ 0.- Phần tử đầu tiên trong mảng có chỉ số là 0.- Phần tử cuối cùng có chỉ số là <số phần tử mảng -1>.	<pre>int test[5]; //khai báo mảng test có tối đa 5 phần tử, mỗi phần tử có kiểu <i>int</i>. //phần tử đầu tiên có chỉ số là 0 //phần tử cuối cùng có chỉ số là 4.</pre>

Bài 7: MẢNG MỘT CHIỀU

<ul style="list-style-type: none">Truy xuất các phần tử mảng theo cú pháp: <tên mảng>[chỉ số phần tử]Nhập giá trị cho các phần tử mảng sử dụng đối tượng cinXuất giá trị phần tử mảng, sử dụng đối tượng cout	<pre>//phần tử đầu tiên test[0] // phần tử thứ hai. test[1] //phần tử cuối cùng. test[4] cin>>test[0]; //nhập giá trị cho phần tử đầu tiên. cout<<test[4]; //xuất giá trị phần tử cuối cùng trong mảng test ra màn hình</pre>
<ul style="list-style-type: none">Nhập giá trị cho mảng sử dụng <u>một vòng lặp</u> duyệt mảng <pre>for(int i=0;i<sophantumang;i++){ cout<<"\nNhap phan tu:"<<i<<": "; cin>>TenMang[i]; }</pre> <ul style="list-style-type: none">Xuất giá trị của mảng sử dụng <u>một vòng lặp</u> duyệt mảng	<pre>int a[10]; for(int i=0;i<10;i++){ cout<<"Ptu"<<i<<": "; cin>>a[i]; cout<<endl; } //vói mảng a đã được //khai báo ở trên</pre>

```
for (int  
i=0; i<sophantumang; i++)  
  
    cout<<TenMang[i]<<"\t";
```

```
for (int  
i=0; i<10; i++)  
  
    cout<<a[i]<<"\t";
```

III. NỘI DUNG THỰC HÀNH

Tài nguyên: D:\NMLT\Lab7\Examples

1. Chương trình mẫu (0.4 điểm)

a) Đây là đoạn mã lệnh chương trình được lưu trong tập tin **lab7_a.cpp**. Yêu cầu của đoạn chương trình là cho phép:

- Nhập vào một mảng gồm 10 phần tử , mỗi phần tử là số nguyên.
- Xuất mảng vừa nhập ra màn hình.
- Xuất các phần tử là số dương có trong mảng.

Yêu cầu bài toán:

- Xây dựng hàm **nhap_Mang** để nhập giá trị cho số phần tử của mảng.
- Xây dựng hàm **xuat_Mang** để xuất các phần tử mảng ra màn hình.
- Xây dựng hàm **xuat_SoDương** để xuất các phần tử là số dương.
 - Xây dựng menu để lựa chọn các chức năng của chương trình.
 - Viết hàm main gọi lại các hàm trên

Đây là toàn bộ mã lệnh của chương trình.

- File thư viện như sau :

Mã lệnh

```
#include<iostream>
using namespace std;
void nhap_Mang(int b[]);
void xuat_Mang(int b[]);
void xuat_SoDuong(int b[]);
void hienThi_Menu();
void chon_Menu(int b[]);
```

- File cài đặt như sau :

Mã lệnh

```
#include"thuvien_lab_7.h"
void nhap_Mang(int b[]){
    int i;
    cout << "nNhap gia tri cho mang:";
    for (i = 0; i<10; i++){
        cout << "\nNhap phan tu thu :" << i << ":
";
        cin >> b[i];
    }
}
void xuat_Mang(int b[]){
    cout << "\nCac phan tu mang la: ";
    for (int i = 0; i<10; i++)
        cout << b[i] << "\t";
    cout << endl;
}
/*
ham xuat cac so duong co trong mang.
duyet lan luot cac phan tu cua mang
neu phan tu nao la so duong thi xuat ra man hinh
*/
void xuat_SoDuong(int b[]){
    for (int i = 0; i<10; i++)
        if (b[i]>0)//kiem tra dieu kien duong
            cout << b[i] << "\t";
    cout << endl;
}
```

```
/*Hien thi cac chuc nang cua chuong trinh*/
void hienThi_Menu(){
    cout << endl;
    cout << "\n===== ";
    cout << "\n Cac chuc nang chuong trinh." << endl;
    cout << "\n1. Nhap mang so nguyen." << endl;
    cout << "\n2. Xuat cac phan tu cua mang." <<
endl;
    cout << "\n3. Xuat cac phan tu duong cua mang."
<< endl;
    cout << "\n0. Dung chuong trinh." << endl;
    cout << endl;
    cout << "\n===== ";
}
/*
Ham lua chon cac chuc nang cua chuong trinh
*/
void chon_Menu(int b[]){
    int chon;
    do{
        cout << "\nNhap 1 so nguyen de lua chon
chuc nang...";
        cin >> chon;
        switch (chon){
            case 1:
                //xoa noi dung dang hien thi tren man
hinh
                system("cls");
                nhap_Mang(b);
                hienThi_Menu();
                break;
            case 2:
                system("cls");
                xuat_Mang(b);
                hienThi_Menu();
                break;
            case 3:
                system("cls");
```

Bài 7: MẢNG MỘT CHIỀU

```
        cout << "\nCac so duong co trong mang  
la:" << endl;  
        xuat_SoDuong(b);  
        hienThi_Menu();  
        break;  
    default:  
        break;  
    }  
} while (chon != 0);  
}
```

- File chương trình như sau :

Mã lệnh

```
#include "thuvien_lab_7.h"  
void main(){  
    int a[10];  
    hienThi_Menu();  
    chon_Menu(a);  
    system("pause");  
}
```

Yêu cầu sinh viên:

a.1/ Đọc và giải thích hàm **nhap_Mang**.

a.2/ Mảng được sử dụng là tham số của hàm **nhap_Mang** được sử dụng theo cơ chế truyền tham trị hay tham biến?

a.3/ Viết giải thích cho hàm **xuat_SoDuong**.

a.4/ Hãy sửa lại dòng lệnh tương ứng để xuất ra màn hình các số lẻ.

b) Cho đoạn mã sau:

- Tập tin thư viện như sau:

Mã lệnh

Bài 7: MẢNG MỘT CHIỀU

```
#include<iostream>
using namespace std;
const int KTM = 50;
//khai bao cac ham con
void nhap_SoPhanTu(int &);
void nhap_Mang(int b[], int m);
void xuat_Mang(int b[], int m);
int kiemTra_HoanThien(int k);
void xuat_SoHoanThien(int b[], int m);
int tong_PTuDuong(int b[], int m);
int tim_Max(int b[], int m);
void hienThi_Menu();
void chon_Menu(int b[], int m);
```

- Tập tin cài đặt như sau:

Mã lệnh

```
#include"thuvien_lab7.h"
void nhap_SoPhanTu(int &m){
    do{
        cout << "\nNhap so phan tu cua mang...";
        cin >> m;
    } while (m <= 0 || m>KTM);
}
//nhap mang b co m phan tu
void nhap_Mang(int b[], int m){
    int i;
    cout << "nNhap gia tri cho mang:";
    for (i = 0; i<m; i++){
        cout << "\nNhap phan tu thu :" << i << ":
";
        cin >> b[i];
    }
}
void xuat_Mang(int b[], int m){
    cout << "\nCac phan tu mang la:" << endl;
    for (int i = 0; i<m; i++)
        cout << b[i] << "\t";
    cout << endl;
```

```
}
int kiemTra_HoanThien(int k){
    int sum = 0;
    for (int i = 1; i<k; i++)
        if (k%i == 0)
            sum += i;
    if (sum == k)
        return 1;
    return 0;
}
/*
ham xuất các số hoàn thiện có trong mảng.
duyet lan luot các phần tử của mảng
neu phần tử nào là số hoàn thiện thì xuất ra màn hình
*/
void xuất_SoHoanThien(int b[], int m){
    for (int i = 0; i<10; i++)
        if (kt_HoanThien(b[i]) == 1)
            cout << b[i] << "\t";
    cout << endl;
}
int tong_PTuDuong(int b[], int m){
    int sum = 0;
    for (int i = 0; i<m; i++)
        if (b[i]>0)
            sum += b[i];
    return sum;
}
int tim_Max(int b[], int m){
    int max = b[0];
    for (int i = 1; i<m; i++)
        if (max<b[i])
            max = b[i];
    return max;
}
/*Hien thi các chức năng của chương trình*/
void hienThi_Menu()
{
```

```
    cout << "\n=====";
    cout << "\nCác chức năng chương trình." << endl;
    cout << "\n1. Nhập mảng số nguyên." << endl;
    cout << "\n2. Xuất các phần tử của mảng." <<
endl;
    cout << "\n3. Các phần tử là số hoán thiên của
mảng." << endl;
    cout << "\n4. Tính tổng các phần tử dương có
trong mảng." << endl;
    cout << "\n5. Giá trị lớn nhất của mảng." <<
endl;
    cout << "\n0. Dung chương trình." << endl;
    cout << "\n=====";
}
/*
Ham lua chon cac chuc nang cua chương trình
*/
void chon_Menu(int b[], int m)
{
    int chon;
    do{
        cout << "\nNhập 1 số nguyên để lựa chọn chức
năng...";
        cin >> chon;
        switch (chon)
        {
            case 1:
                //xóa nội dung đang hiển thị trên màn
hình
                system("cls");
                nhập_Mảng(b, m);
                hiểnThị_Menu();
                break;
            case 2:
                system("cls");
                xuất_Mảng(b, m);
        }
    }
}
```

```
        hienThi_Menu();
        break;
    case 3:
        system("cls");
        cout << "\nCac hoan thien co trong mang
la:" << endl;
        xuat_SoHoanThien(b, m);
        hienThi_Menu();
        break;
    case 4:
        system("cls");
        cout << "\nTong cac so duong co trong
mang la:";
        cout << tong_PTuDuong(b, m) << endl;
        hienThi_Menu();
        break;
    case 5:
        system("cls");
        cout << "\nGia tri lon nhat la:" <<
tim_Max(b, m);
        hienThi_Menu();
        break;
    default:
        break;
    }
} while (chon != 0);
}
```

- Tập tin chương trình:

Mã lệnh

```
#include "thuvien_lab7.h"
void main(){
    //khai bao mang so nguyen co toi da 50 phan tu
    int a[KTM];
    int n; //so phan tu cua mang
    //nhap so phan tu mang
    nhap_SoPhanTu(n);
```

```
hienThi_Menu();  
chon_Menu(a, n);  
system("pause");  
}
```

Yêu cầu sinh viên:

b.1/ Hãy viết tóm tắt lại các yêu cầu của bài toán.

b.2/ Trong hàm tổng, hãy sửa lại dòng lệnh tương ứng để thực hiện tính tổng các phần tử là số dương mà không là bội số của 4.

2. Chỉnh sửa số liệu theo yêu cầu (0.2 điểm)

- Sửa lại hàm **tong_PtuDuong** để tính tổng các phần tử là số nguyên âm có trong mảng.
- Xây dựng thêm hàm mới **tim_Min** tìm phần tử có giá trị nhỏ nhất trong mảng.

Hướng dẫn: giống hàm **tim_Max** trong **lab7_b.cpp** nhưng thay đổi điều kiện so sánh.

- Xây dựng thêm hàm mới **dem_so_PtuAm** trả về số phần tử có giá trị âm trong mảng.

Ví dụ: mảng có các phần tử: 3 -2 5 6 7 -9 5. Số phần tử âm có trong mảng là 2.

- Xây dựng hàm **sochiahet3** trả về số lượng các phần tử chia hết cho 3 có trong mảng.

Ví dụ: mảng có các phần tử: 3 -2 5 6 7 -9 5. Số phần tử chia hết cho 3 có trong mảng là 3.

3. Bài tập vận dụng (0.2 điểm)

Viết chương trình cho phép nhập vào một mảng gồm có n phần tử là số

nguyên. Hãy viết các hàm thực hiện các công việc sau:

- Xuất giá trị các phần tử mảng vừa nhập ra màn hình.
- Xuất các phần tử là số chẵn ra màn hình.
- Hàm **tong_BoiSo7** để tính tổng các phần tử là số nguyên dương và chia hết cho 7 có trong mảng.
- Hàm **TBCong** để tính trung bình cộng các số nguyên dương có trong mảng.

4. Bài tập tổng hợp (0.2 điểm)

Viết chương trình trên giấy (không sử dụng máy tính) bài toán sau:

Xây dựng các hàm thực hiện các công việc sau:

- Xây dựng hàm **nhap_Mang** để nhập giá trị mảng.
- Xây dựng hàm **xuat_Mang** để xuất mảng.
- Xây dựng hàm **kt_SoNguyenTo** để kiểm tra một số có phải là số nguyên tố không?
- Xây dựng hàm **xuat_SoNguyenTo** để xuất các phần tử là số nguyên tố.
- Xây dựng hàm **tong_SoNguyenTo** để tính tổng các phần tử là số nguyên tố.
- Xây dựng hàm **main** gọi lại các hàm trên.

IV. BÀI TẬP LÀM THÊM

1. Viết chương trình nhập mảng các số thực và xuất các phần tử âm trong mảng.

Bài 7: MẢNG MỘT CHIỀU

2. Viết chương trình nhập mảng các số nguyên và xuất các phần tử lẻ có trong mảng.
3. Viết chương trình nhập vào mảng một chiều các số nguyên và xuất ra các phần tử chẵn nhỏ hơn 20.
4. Viết chương trình nhập vào mảng một chiều các số nguyên và xuất ra màn hình các phần tử là số nguyên tố.
5. Viết chương trình nhập vào mảng một chiều các số nguyên và xuất ra màn hình các phần tử là số chính phương nằm tại những vị trí lẻ trong mảng.

BÀI 8: MẢNG HAI CHIỀU**I. MỤC TIÊU**

Sau khi thực hành xong bài này, sinh viên phải biết cách:

- Định nghĩa và thao tác trên mảng hai chiều.

II. TÓM TẮT LÝ THUYẾT

Nội dung	Ví dụ
<ul style="list-style-type: none">▪ Mảng hai chiều tương tự như một bảng tính Excel.▪ Khai báo mảng hai chiều: <i><kiểu dữ liệu> <tên mảng>[số dòng][số cột];</i> <p>+ Số phần tử của mảng là: [số dòng] * [số cột]</p> <p>+ Phần tử đầu tiên có chỉ số là [0][0]</p> <p>+ Phần tử cuối cùng có chỉ số là: [số dòng -1][số cột -1].</p> <p>+ Truy xuất các phần tử mảng theo cú pháp: <i><tên mảng>[<chỉ số dòng>][<chỉ số cột>]</i></p>	<pre>int test[3][4]; //khai báo mảng test có 3 dòng 4 cột, có 12 phần tử, mỗi phần tử có kiểu int. x = test[0][0]; //gán cho x giá trị của phần tử dòng 0, cột 0 trong</pre>

	<p>mảng test.</p> <pre>test[2][3]=test[1][0];</pre> <p>// lấy giá trị của phần tử dòng 1 cột 0 trong mảng test gán cho phần tử dòng 2, cột 3 trong mảng test.</p>
<ul style="list-style-type: none"> Sử dụng đối tượng cin để nhập giá trị cho từng phần tử mảng, cout để xuất. 	<pre>cin>>test[0][0];</pre> <p>//nhập giá trị cho phần tử dòng 0 cột 0.</p> <pre>cout<<test[1][2];</pre> <p>//xuất giá trị của phần tử dòng 1 cột 2 ra màn hình.</p>
<ul style="list-style-type: none"> Nhập giá trị cho mảng hai chiều: <pre>for(int i=0;i<số dòng;i++) for(int j=0;j<số cột;j++){ cout<<"\nPt["<i<<"["<j<<":"; ; cin>>tenmang[i][j]; }</pre> <ul style="list-style-type: none"> Xuất giá trị của mảng hai chiều: 	<p>//Nhập giá trị cho mảng hai chiều có tên là a có số dòng là n, số cột là m:</p> <pre>for(int i=0;i<n;i++) for(int j=0;j<m;j++){ cout<<"\nPt["<i<<["<j<<":";</pre>

Bài 8: MẢNG HAI CHIỀU

```
for(int i=0;i<số dòng;i++){
    for(int j=0;j<số cột;j++){
        cout<<tenmang[i][j]<<"\t
    ";
    cout<<endl;
}
```

```
cin>>a[i][j];
}

//Xuất giá trị của mảng
hai chiều có tên là a, có
số dòng là n, số cột là m:

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

        cout<<a[i][j]
        <<"\t";
    cout<<endl;
}
```

III. NỘI DUNG THỰC HÀNH

Tài nguyên: D:\NMLT\Lab8\Examples

1. Chương trình mẫu (0.4 điểm)

a) Đây là đoạn mã lệnh chương trình được lưu trong tập tin *lab8_a.cpp*. Chương trình được tổ chức dưới dạng chương trình con, cho phép:

- Nhập vào một ma trận gồm 2 dòng 3 cột, mỗi phần tử là số nguyên.
 - Xuất toàn bộ ma trận vừa nhập ra màn hình.
 - Tính tổng các phần tử có trong ma trận.
- Tập tin thư viện như sau:

Mã lệnh

```
#include<iostream>
using namespace std;
void nhap_MaTran(int b[][3]);
void xuat_MaTran(int b[][3]);
```

```
int tinhTong_MaTran(int b[][3]);
```

- Tập tin cài đặt:

Mã lệnh

```
#include"thuvien_lab8.h"
//viet dinh nghĩa ham nhap ma tran
void nhap_MaTran(int b[][3]){
    cout << "\nNhap gia tri cho cac phan tu" << endl;
    for (int i = 0; i<2; i++)//?
        for (int j = 0; j<3; j++){
            cout << "\nNhap phan tu " << "[" << i <<
            "]"[" << j << "]:";
            cin >> b[i][j];
        }
}
//viet dinh nghĩa ham xuat ma tran
void xuat_MaTran(int b[][3]){
    cout << endl;
    for (int i = 0; i<2; i++){
        for (int j = 0; j<3; j++)
            cout << b[i][j] << "\t";
        cout << endl;//?
    }
}
//viet dinh nghĩa ham tong ma tran
int tinhTong_MaTran(int b[][3]){
    int sum = 0;
    for (int i = 0; i<2; i++)
        for (int j = 0; j<3; j++)
            sum += b[i][j];//?
    return sum;
}
```

- Tập tin chương trình

Mã lệnh

```
#include "thuvien_lab8.h"
void main(){
    //khai bao ma tran 2 dong 3 cot
    int a[2][3];
    cout << "\nNhap ma tran:" << endl;
    nhap_MaTran(a);
    cout << "\nMa tran nhap vao la:" << endl;
    xuat_MaTran(a);
    cout << "\nTong cac phan tu ma tran :" <<
    tinhTong_MaTran(a) << endl;
    system("pause");
}
```

Yêu cầu sinh viên:

- a.1/ Đọc mã lệnh chương trình trong các hàm: **nhap_MaTran**, **xuat_MaTran**, **tong_MaTran**
 - a.2/ Ma trận sử dụng trong chương trình là ma trận số nguyên. Hãy sửa lại dòng lệnh tương ứng để cho phép nhập vào một ma trận các **số thực**.
 - a.3/ Ma trận nhập vào là ma trận có hai dòng và ba cột. Hãy sửa lại dòng lệnh tương ứng để nhập vào một ma trận có **ba dòng** và **bốn cột**.
 - a.4/ Ta có thể viết lại **nhap_MaTran(a[2][3])** được hay không? Giải thích.
 - a.5/ Ở **void main** ta có thể viết lại **nhap_MaTran(b)** được hay không? Giải thích.
- b) Đây là đoạn mã lệnh chương trình nội dung như sau:

Bài 8: MẢNG HAI CHIỀU

- File thư viện như sau:

Mã lệnh

```
#include<iostream>
using namespace std;
const int KTMT = 10;
//khai bao cac ham con
void nhap_1SoNguyen(int &k);
void nhap_MaTran(int b[][KTMT], int m);
void xuat_MaTran(int b[][KTMT], int m);
int tong_DuongCheoPhu(int b[][KTMT], int m);
void xuat_BoiSo5(int b[][KTMT], int m);
int tong_SoLe(int b[][KTMT], int m);
```

- File cài đặt như sau:

Mã lệnh

```
#include"thuvien_lab8.h"
//viet dinh nghĩa các hàm con
void nhap_1SoNguyen(int &k){
    do{
        cout << "\nNhap 1 so : ";
        cin >> k;
    } while (k <= 0 || k>KTMT);
}
void nhap_MaTran(int b[][KTMT], int m){
    for (int i = 0; i<m; i++)
        for (int j = 0; j<m; j++){
            cout << "\nNhap PTu[" << i << "][" << j <<
": ";
            cin >> b[i][j];
        }
}
void xuat_MaTran(int b[][KTMT], int m)
{
    cout << endl;
```

```
for (int i = 0; i<m; i++)
{
    for (int j = 0; j<m; j++)
        cout << b[i][j] << "\t";
    cout << endl;
}
}
/*
Phan tu thuoc duong cheo phu la phan tu
co chi so dong + chi so cot = so dong ma tran -1
vi du: matran a co 4 dong 4 cot thi cac phan tu phu
la
a[0][3],a[1][2],a[2][1],a[4][0]
*/
int tong_DuongCheoPhu(int b[][KTMT], int m)
{
    int s = 0;
    for (int i = 0; i<m; i++)
        s += b[i][m - 1 - i];
    return s;
}
void xuat_BoiSo5(int b[][KTMT], int m){
    cout << endl;
    for (int i = 0; i<m; i++){
        for (int j = 0; j<m; j++)
            if (b[i][j] % 5 == 0)
                cout << b[i][j] << "\t";
            else
                cout << "*" << "\t";
        cout << endl;
    }
}
int tong_SoLe(int b[][KTMT], int m){
    int sum = 0;
    for (int i = 0; i<m; i++)
        for (int j = 0; j<m; j++)
            if (b[i][j] % 2 == 0)
                sum += b[i][j];
}
```

Bài 8: MẢNG HAI CHIỀU

```
    return sum;
}
```

- File chương trình:

Mã lệnh

```
#include "thuvien_lab8.h"
void main(){
    int a[KTMT][KTMT];
    //khai bao so dong , so cot ma tran
    int n;//m kích thước ma tran vuong
    cout << "\nNhap kích thước ma tran :";
    nhap_1SoNguyen(n);
    cout << "\nNhap ma tran : " << endl;
    nhap_MaTran(a, n);
    cout << "\nMa tran nhập vào là : " << endl;
    xuất_MaTran(a, n);
    cout << "\nTổng các phần tử trên đường chéo
phu:";
    cout << tong_DuongCheoPhu(a, n);
    cout << "\nCác phần tử là bội số của 5 là : " <<
endl;
    xuất_BoiSo5(a, n);
    cout << "\nTổng các số lẻ có trong ma tran là :
";
    cout << tong_SoLe(a, n) << endl;
    system("pause");
}
```

Yêu cầu sinh viên:

- b.1/** Viết tóm tắt lại các yêu cầu của bài toán.
- b.2/** Hãy sửa lại dòng lệnh tương ứng trong hàm **tong_SoLe** để tính tổng các phần tử là số nguyên dương và là bội số của 3.

b.3/ Nếu muốn ma trận nhập vào là ma trận các số thực thì phải chỉnh sửa dòng lệnh nào? Hãy viết lại dòng lệnh đã chỉnh sửa.

2. Chỉnh sửa số liệu theo yêu cầu (0.2 điểm)

Mở **Lab8_b.cpp**, và thực hiện các công việc sau:

- Sửa lại hàm **tong_SoLe** để thực hiện tính tổng các số dương lẻ có trong ma trận.
- Xây dựng hàm **in_DuongCheoChinh** để in ra các phần tử nằm trên đường chéo chính của ma trận. (phần tử trên đường chéo chính là phần tử có chỉ số dòng bằng chỉ số cột). Gợi ý: dựa vào hàm **tong_DuongCheoPhu**
- Sửa lại hàm **xuat_BoiSo5** để xuất các phần tử là vừa chia hết cho 3 và 7.

3. Bài tập vận dụng (0.2 điểm)

Viết chương trình cho phép nhập vào một ma trận vuông có kích thước $n \times n$, mỗi phần tử là số nguyên. Sau đó thực hiện các công việc sau:

- Xuất ma trận vừa nhập ra màn hình.
- Xuất các phần tử là số lẻ ra màn hình, các phần tử không lẻ thay bằng kí tự *.
- Tính tổng các phần tử là số nguyên dương và là bội số của 3 và 5 có trong ma trận.
- Tính trung bình cộng các số nguyên dương có trong ma trận.

Yêu cầu: viết các yêu cầu trên dưới dạng hàm.

4. Bài tập tổng hợp (0.2 điểm)

Viết chương trình trên giấy (không sử dụng máy tính) các bài toán sau:

Viết chương trình cho phép nhập vào một ma trận có kích thước $n \times n$, mỗi phần tử là một số nguyên. Sau đó thực hiện các công việc sau:

- Xuất ma trận vừa nhập ra màn hình.
- Xuất các phần tử là số nguyên tố có trong ma trận ra màn hình.
- Tính tổng các phần tử là số nguyên tố có trong ma trận.
- Tính tổng các phần tử trên dòng k của ma trận.

Yêu cầu:

- Xây dựng hàm **nhap_MaTran** để nhập ma trận
- Xây dựng hàm **xuat_MaTran** để xuất ma trận.
- Xây dựng hàm **kt_SoNguyenTo** để kiểm tra một số có phải là số nguyên tố không?
- Xây dựng hàm **xuat_SoNguyenTo** để xuất các phần tử là số nguyên tố.
- Xây dựng hàm **tong_SoNguyenTo** để tính tổng các phần tử là số nguyên tố.
- Xây dựng hàm **tong_Dongk** để tính tổng các phần tử trên dòng k .

IV. BÀI TẬP LÀM THÊM

1. Viết hàm đếm các phần tử âm, hàm đếm các phần tử dương trong ma trận.
2. Viết hàm đếm các phần tử chẵn, hàm đếm các phần tử lẻ trong ma trận.
3. Viết hàm đếm số lần xuất hiện của phần tử x trong ma trận.

Bài 8: MẢNG HAI CHIỀU

4. Viết hàm đếm các phần tử nhỏ hơn x trong ma trận.
5. Viết hàm đếm các phần tử là số nguyên tố trong ma trận.
6. Viết hàm đếm các phần tử là số hoàn thiện trong ma trận.
7. Viết hàm đếm các phần tử là bội của 3 hoặc 5 trong ma trận các số nguyên.
8. Viết hàm tính tổng các phần tử nằm ở dòng chẵn trong ma trận các số nguyên.

BÀI 9: XÂY KÍ TỰ**I. MỤC TIÊU**

Sau khi thực hành xong bài này, sinh viên phải biết cách:

- Khai báo và thao tác trên xâu kí tự.

II. TÓM TẮT LÝ THUYẾT

Nội dung	Ví dụ
<ul style="list-style-type: none">▪ Xâu kí tự có cấu trúc mảng.▪ Khai báo: <i>char <tênbiến><[kích thước chuỗi]>;</i>▪ Kích thước xâu kí tự: là số kí tự tối đa nhập vào cho biến +1.▪ Một xâu kí tự được kết thúc bằng kí tự ‘\0’▪ Truy xuất các kí tự trong xâu theo cú pháp: <i><tên biến>[vị trí kí tự]</i>	<pre>char line[40]; //khai báo biến line có thể chứa tối đa 39 kí tự. //phần tử đầu tiên có chỉ số là [0], phần tử cuối cùng có chỉ số là [38]. + line[0] = 'A'; //kí tự đầu tiên trong xâu line được gán là A + line[10] // kí tự thứ 10 trong xâu line</pre>
<p>Lưu ý:</p> <ul style="list-style-type: none">▪ Sử dụng đối tượng cin sẽ	<pre>gets(line);</pre>

<p>nhập xâu kí tự không chứa kí tự trắng.</p> <ul style="list-style-type: none">▪ Sử dụng hàm gets(<tên biến xâu kí tự>) hoặc cin.getline(tên biến, số kí tự) nhập giá trị cho xâu kí tự, có thể nhập cả kí tự trắng.	<p>//nhập giá trị cho biến line và khai báo ở trên.</p> <pre>cin.getline(line, 40) ;</pre>
---	---

Một số hàm thao tác trên xâu kí tự (khai báo thư viện *string.h*):

- **strlen():** xác định chiều dài của xâu kí tự.
- **strcat(string1,string2):** nối xâu string2 vào cuối xâu string1.
- **strcpy(string1,string2):** copy xâu string2 lên xâu string1.
- **strcmp(string1,string2):** so sánh xâu string2 và xâu string 1.

Nếu kết quả hàm trả về:

- 0 nếu string1 = string2.
- <0 nếu string1 < string2.
- >0 nếu string1 > string2.

Một số hàm trong thư viện *cctype.h*

- **toupper(charc):** chuyển kí tự charc về kí tự in hoa.
- **tolower(charc):** chuyển kí tự charc về kí tự thường.

III. NỘI DUNG THỰC HÀNH

Tài nguyên: D:\NMLT\Lab9\Examples

1. Chương trình mẫu (0.4 điểm)

a) Đây là đoạn mã lệnh chương trình được lưu trong tập tin *lab9_a.cpp*, cho phép:

- Nhập vào một chuỗi.
 - Cắt bỏ các khoảng trắng dư thừa trong chuỗi (đầu chuỗi, cuối chuỗi và những khoảng trắng giữa chuỗi).
 - Chuyển tất cả các kí tự có mặt trong chuỗi thành kí tự in hoa.
- File thư viện như sau :

Mã lệnh
<pre>#include<iostream> #include<string> #include<cctype> using namespace std; void chuanHoa_Chuoai(char chuoai[]); void chuyen_Chuhua(char chuoai[]);</pre>

- File cài đặt như sau :

Mã lệnh
<pre>#include"thuvien_lab9.h" void chuyen_Chuhua(char chuoai[]){ for (unsigned int i = 0; i<strlen(chuoai); i++) chuoai[i] = toupper(chuoai[i]); //? } //hoan thien cac ham con tai day void chuanHoa_Chuoai(char chuoai[]){ unsigned int i; //cat bo nhung khoang trang dau chuoai while (chuoai[0] == ' '){ for (unsigned int j = 1; j<strlen(chuoai); j++) chuoai[j - 1] = chuoai[j]; //?</pre>

```
        chuoai[strlen(chuoai) - 1] = '\\0';
    //?
    }
    //cat bo nhung khoang trang cuoi chuoai
    while (chuoai[strlen(chuoai) - 1] == ' '){
        chuoai[strlen(chuoai) - 1] = '\\0';
    }
    //cat bo nhung khoang trang o giua chuoai
    for (i = 0; i<strlen(chuoai); i++)
        if (chuoai[i] == ' '){
            while (chuoai[i + 1] == ' '){
                //tinh tien chuoai phia sau len
                for (unsigned int j = i + 1;
j<strlen(chuoai); j++)
                    chuoai[j] = chuoai[j+ 1];
                chuoai[strlen(chuoai)] = '\\0';
            }
        }
}
```

- File chương trình như sau :

Mã lệnh

```
#include"thuvien_lab9.h"
void main(){
    char str[100];
    cout << "\\nNhap vao mot chuoai...";
    cin.getline(str, 100);
    cout << "\\nChuoai nhap vao la:" << str;
    chuanHoa_Chuoai(str);
    cout << "\\nChuoai sau khi cat khoang trang
la:#" << str << "#";
    chuyen_Chuhoa(str);
    cout << "\\nChuoai hoa la:" << str << endl;
    system("pause");
}
```

Yêu cầu sinh viên:

a.1/ Thực thi **Lab9_a.cpp**.

a.2/ Trong hàm **chuanhoa_Chuoi**, nếu bỏ các dòng lệnh **chuoi[strlen(chuoi)-1]='\0'**; thì chương trình có hoạt động được hay không? Giải thích tại sao?

a.3/ Hãy sửa lại dòng lệnh tương ứng trong hàm **chuyen_Chuhoa** để chuyển các kí tự là kí tự a có mặt trong chuỗi thành chữ in hoa, còn các kí tự khác chuyển thành kí tự thường. Ví dụ: Ban mai => bAn mAi.

b) Đây là đoạn mã lệnh chương trình được lưu trong tập tin **lab9_b.cpp**, nội dung đoạn mã lệnh như sau:

- File thư viện như sau:

Mã lệnh
<pre>#include<iostream> #include<string> #include<cctype> using namespace std; void chuanHoa_Chuoi(char chuoi[]); int tim_Kitu(char chuoi[], char c); int kt_ChuoiDoixung(char chuoi[]); void chuyen_kituM_hoa(char chuoi[]); void chuyen_ChuThuong(char chuoi[]);</pre>

- File cài đặt như sau:

Mã lệnh
<pre>#include"thuvien_lab9.h" void chuyen_Chuhoa(char chuoi[]){ for (unsigned int i = 0; i<strlen(chuoi); i++) chuoi[i] = toupper(chuoi[i]); //? } //hoan thien cac ham con tai day</pre>

```
void chuanHoa_Chuoai(char chuoai[]){
    unsigned int i;
    //cat bo nhung khoang trang dau chuoai
    while (chuoai[0] == ' '){
        for (unsigned int j = 1; j<strlen(chuoai);
j++)
            chuoai[j - 1] = chuoai[j];
        //?
        chuoai[strlen(chuoai) - 1] = '\0';
        //?
    }
    //cat bo nhung khoang trang cuoi chuoai
    while (chuoai[strlen(chuoai) - 1] == ' '){
        chuoai[strlen(chuoai) - 1] = '\0';
    }
    //cat bo nhung khoang trang o giua chuoai
    for (i = 0; i<strlen(chuoai); i++)
        if (chuoai[i] == ' '){
            while (chuoai[i + 1] == ' '){
                //tinh tien chuoai phia sau len
                for (unsigned int j = i + 1;
j<strlen(chuoai); j++)
                    chuoai[j] = chuoai[j + 1];
                chuoai[strlen(chuoai)] = '\0';
            }
        }
    //dem so ky tu co trong chuoai
    int tim_Kitu(char chuoai[], char c)
    {
        int dem = 0;
        for (int i = 0; i<strlen(chuoai); i++)
            if (chuoai[i] == c)
                dem++;
        return dem;
    }
    //kiem tra chuoai doi xung; co tra ve 1; ko tra ve 0
    int kt_ChuoaiDoixung(char chuoai[])
    {
```

Bài 9: XÂU KÍ TỰ

```
int len = strlen(chuoi);
int i = 0;
while (chuoi[i] == chuoi[len - i - 1] && i < len
/ 2)
    i++;
if (i == len / 2)
    return 1;
else
    return 0;
}
void chuyen_kituM_hoa(char chuoi[]){
    for (int i = 0; i < strlen(chuoi); i++)
        if (chuoi[i] == 'm')
            chuoi[i] = toupper(chuoi[i]);
        else
            chuoi[i] = tolower(chuoi[i]);
}
void chuyen_ChuThuong(char chuoi[]){
    for (int i = 0; i < strlen(chuoi); i++)
        chuoi[i] = tolower(chuoi[i]);
}
```

- File chương trình:

Mã lệnh

```
#include "thuvien_lab9.h"
void main(){
    char str[100];
    cout << "\nNhap vao mot chuoi...";
    cin.getline(str, 100);
    cout << "\nChuoi nhap vao la:" << str;
    chuanHoa_Chuoi(str);
    cout << "\nChuoi sau khi cat khoang trang la:#"
<< str << "#";
    chuyen_ChuThuong(str);
    cout << "\nChuoi ki tu thuong la:" << str <<
endl;
```

```
chuyen_kituM_hoa(str);
cout << "\nChuyen tat ca ki tu M thanh in hoa
la:" << endl;
cout << str << endl;
cout << "\nSo ki tu N co trong chuoai la: " <<
tim_Kitu(str, 'N');
if (kt_ChuoaiDoixung(str))
    cout << "\nchuoai doi xung\n";
else
    cout << "\nChuoai khong doi xung\n";

system("pause");
}
```

Yêu cầu sinh viên:

b.1/ Hãy nhập đoạn mã trên

b.2/ Viết lại các yêu cầu bài toán.

b.3/ Hãy chỉnh sửa dòng lệnh tương ứng ở hàm `chuyen_kituM_hoa` để chỉ chuyển các kí tự đứng trước kí tự trắng thành kí tự in hoa. Ví dụ: chuỗi ban đầu như sau: **truong dai hoc cong nghe sai gon**. Sau khi gọi hàm `M_hoa` sẽ chuyển thành chuỗi sau: **truonG daI hoC conG nghE sai gon**.

2. Chỉnh sửa số liệu theo yêu cầu (0.2 điểm)

Mở **Lab9_a.cpp**, hãy sửa lại các hàm để thực hiện các công việc sau:

- Nhập vào một chuỗi.
- Đếm số từ có trong chuỗi.

Gợi ý: số từ chính là số khoảng trắng có trong chuỗi sau khi đã cắt bỏ các khoảng trắng dư thừa cộng thêm 1).

- Đổi các kí tự đầu của mỗi từ thành kí tự in hoa.

Ví dụ: trường đại học công nghệ sài gòn -> Trường Đại Học Công Nghệ Sài Gòn

3. Bài tập vận dụng (0.2 điểm)

Mở Lab9_b.cpp, hãy sửa lại các hàm để thực hiện các công việc sau:

- Nhập vào một chuỗi.
- Tìm số lần xuất hiện của một kí tự bất kỳ được nhập từ bàn phím trong chuỗi.
- Đổi các kí tự đầu của mỗi từ thành kí tự thường và các kí tự còn lại thành kí tự in hoa.

Ví dụ: lam viec -> lAM vIEC

4. Bài tập tổng hợp (0.2 điểm)

Viết chương trình trên giấy (không sử dụng máy tính) các bài toán sau:

Hãy xây dựng các hàm cho phép thực hiện các công việc sau:

- Nhập vào một chuỗi.
- Đếm số lần xuất hiện của kí tự ‘a’ hoặc ‘A’ có trong chuỗi.
- Đổi kí tự hoa thành kí tự thường và kí tự thường thành kí tự in hoa. Ví dụ: nếu nhập chuỗi “hoM Nay” => “Hom nAY”
- In chuỗi theo thứ tự đảo ngược. ví dụ: chuỗi nhập vào là: “abcde” thì sẽ in ra “edcba”.

IV. BÀI TẬP LÀM THÊM

1. Viết chương trình nhập vào 1 chuỗi ký tự, cho biết chuỗi có bao nhiêu ký tự ‘a’.
2. Viết chương trình nhập vào 1 chuỗi ký tự, cho biết số lần xuất hiện của các kí tự chữ cái (không tính kí tự số và kí tự đặc biệt) có mặt trong chuỗi.

Ví dụ: hom nay troi mat → h xuất hiện 1 lần

o xuất hiện 2 lần.

m xuất hiện 2 lần.

n xuất hiện 1 lần.

a xuất hiện 1 lần.

y xuất hiện 2 lần.

i xuất hiện 2 lần.

t xuất hiện 2 lần.

Gợi ý: sử dụng một mảng 26 phần tử đại diện cho 26 kí tự.

- Viết chương trình nhập vào 2 chuỗi ký tự str1 và str2. Nếu độ dài của str1 lớn hơn độ dài str2 thì nối chuỗi str2 vào sau chuỗi str1, ngược lại thì nối chuỗi str1 vào sau chuỗi str2.
- Hãy viết chương trình cho phép nhập vào 1 chuỗi, sau đó đảo ngược trật tự các từ có trong chuỗi.

Ví dụ: chuỗi nhập vào là: bo an co

Sẽ đổi lại là: co an bo

Gợi ý: chuỗi nhập vào phải được chuẩn hóa; lần lượt tách các từ trong chuỗi và ghép theo thứ tự ngược lại.

BÀI 10: ÔN TẬP

I. MỤC TIÊU

Lab này tổng hợp kiến thức của các Lab đã thực hiện.

Sau khi thực hiện xong bài thực hành này, sinh viên phải nắm vững các kỹ năng chính sau:

- Xây dựng chương trình con và gọi lại chương trình con.
- Sử dụng tất cả các cấu trúc điều khiển đã học.

Sinh viên nên tập hợp lại lý thuyết tóm tắt của các Lab trước để khi cần có thể tra cứu nhanh.

II. TÓM TẮT LÝ THUYẾT

III. NỘI DUNG THỰC HÀNH

Tài nguyên: D:\NMLT\Lab10\Examples

1. Chương trình mẫu (0.4 điểm)

a) Đây là đoạn mã lệnh chương trình được lưu trong tập tin **lab10_a.cpp**, chương trình cho phép:

Xây dựng một menu chương trình với các chức năng sau:

- 1. Nhập mảng gồm n phần tử là số nguyên.
- 2. Xuất mảng ra màn hình.
- 3. Xuất các phần tử là số nguyên dương có trong mảng.
- 4. Xuất các phần tử là số nguyên chia hết cho 2 và 3.
- 5. Xuất các phần tử là số chẵn có trong mảng.
- 6. Thoát khỏi chương trình.

Bài 10: ÔN TẬP

- File thư viện như sau :

Mã lệnh

```
#include <iostream>
#include "iomanip"
using namespace std;
const int KTM = 50;
void nhap_Mang(int b[], int &m);
void xuất_Mang(int b[], int m);
void xuất_So_Duong(int b[], int m);
void xuất_Chia_2_3(int b[], int m);
void xuất_Chan(int b[], int m);
void hienThi_Menu();
void chọn_Menu(int b[], int &m);
```

- File cài đặt :

Mã lệnh

```
#include "thuvien_lab10.h"
void nhap_Mang(int b[], int &m){
    do{
        cout << "\nNhap vao kích thước mảng:";
        cin >> m;
    } while (m <= 0 || m>KTM);
    cout << "\nNhap giá trị cho các phần tử mảng:";
    for (int i = 0; i<m; i++){
        cout << "\nPhần tử thứ " << i << ": ";
        cin >> b[i];
    }
}
void xuất_Mang(int b[], int m){
    cout << "\nGiá trị các phần tử mảng là:";
    for (int i = 0; i<m; i++)
        cout << setw(5) << b[i];
}
void xuất_So_Duong(int b[], int m){
```

```
    cout << "\nCac phan tu la so duong co trong
mang la:";
    for (int i = 0; i<m; i++)
        if (b[i]>0)
            cout << setw(5) << b[i];
}
void xuat_Chia_2_3(int b[], int m){
    cout << "\nCac phan tu la so chia het cho 2 va
3 la:";
    for (int i = 0; i<m; i++)
        if (b[i] % 2 == 0 && b[i] % 3 == 0)
            cout << setw(5) << b[i];
}
void xuat_Chan(int b[], int m){
    cout << "\nCac phan tu la so chan co trong mang
la:";
    for (int i = 0; i<m; i++)
        if (b[i] % 2 == 0)
            cout << setw(5) << b[i];
}
void hienThi_Menu(){
    cout << "\n\nChon 1 trong 6 so:";
    cout << "\n1.  Nhap mang.";
    cout << "\n2.  Xuat mang.";
    cout << "\n3.  Xuat so nguyen duong.";
    cout << "\n4.  Xuat cac phan tu chia het 2 va
3.";
    cout << "\n5.  Xuat cac phan tu la so chan.";
    cout << "\n0.  Thoat khoi chuong trinh.";
    cout << "\nNhap chon lua:";
}
void chon_Menu(int b[], int &m){
    int chon;
    hienthi_Menu();
    do{
        cout << "\nNhap 1 so de chon chuc
nang...";
        cin >> chon;
```

```
switch (chon)
{
case 1:
    nhap_Mang(b, m);
    hienThi_Menu();
    break;
case 2:
    system("cls");
    xuất_Mang(b, m);
    hienThi_Menu();
    break;
case 3:
    system("cls");
    xuất_So_Duong(b, m);
    hienThi_Menu();
    break;
case 4:
    system("cls");
    xuất_Chia_2_3(b, m);
    hienThi_Menu();
    break;
case 5:
    system("cls");
    xuất_Chan(b, m);
    hienThi_Menu();
    break;
default:
    break;
}
} while (chon != 0);
}
```

- File chương trình :

Mã lệnh

```
#include"thuvien_lab10.h"  
void main(){  
    int a[KTM];  
    int n;  
    chon_Menu(a, n);  
    system("pause");  
}
```

Yêu cầu sinh viên:

- a.1/ Đánh đoạn mã trên để chạy chương trình
- a.2/ Hãy sửa lại dòng lệnh tương ứng trong hàm **xuat_Chan** để xuất ra các phần tử là số chẵn và chia hết cho 5.
- a.3/ Khi thực thi chương trình và nhập vào số 6 thì kết quả nhận được là gì? Nhập 7, kết quả?
- a.4/ Chương trình sẽ ngừng khi nào? Giải thích.

b) Đây là một bài toán với yêu cầu:

Xây dựng một menu chương trình với các chức năng sau:

- 1. Nhập một ma trận vuông kích thước $n \times n$, mỗi phần tử là số nguyên.
 - 2. Xuất ma trận ra màn hình.
 - 3. Xuất các phần tử trên đường chéo phụ của ma trận.
 - 4. Tính tổng các phần tử là số chẵn trong ma trận.
 - 5. Xuất các phần tử là số nguyên tố có trong ma trận.
 - 6. Thoát khỏi chương trình.
- File thư viện như sau:

Mã lệnh

```
#include <iostream>
#include "iomanip"
using namespace std;
const int KTMT = 10;
void nhap_MaTran(int a[][KTMT], int &m);
void xuat_MaTran(int a[][KTMT], int m);
void xuat_Cheo_Phu(int a[][KTMT], int m);
intkiemTra_NTo(int n);
void xuat_NTo(int a[][KTMT], int m);
int tinh_TongChan(int a[][KTMT], int m);
void hienThi_Menu();
void chon_Menu(int a[][KTMT], int &m);
```

- File cài đặt:

Mã lệnh

```
#include "thuvien_lab10.h"
void nhap_MaTran(int a[][KTMT], int &m){
    do{
        cout << "\nNhap vao kich thuoc ma tran
vuong:";
        cin >> m;
    } while (m <= 0 || m>KTMT);
    cout << "\nNhap gia tri cho cac phan tu ma
tran:";
    for (int i = 0; i<m; i++){
        for (int j = 0; j<m; j++){
            cout << "\nPhan tu thu [" << i << "]" <<
j << "]: ";
            cin >> a[i][j];
        }
    }
    void xuat_MaTran(int a[][KTMT], int m){
        cout << "\nGia tri cac phan tu ma tran la:\n";
        for (int i = 0; i<m; i++){
            for (int j = 0; j<m; j++){
                cout << setw(5) << a[i][j];
                cout << endl;
            }
        }
    }
}
```

```
    }  
}  
void xuất_Cheo_Phu(int a[][KTMT], int m){  
    cout << "\nCac phan tu tren duong cheo phu  
la:";  
    for (int i = 0; i<m; i++){  
        cout << setw(5) << a[i][m - i - 1];  
    }  
int kiemTra_NTo(int n){  
    if (n<2)  
        return 0;  
    for (int i = 2; i<n; i++){  
        if (n%i == 0)  
            return 0;  
    }  
    return 1;  
}  
void xuất_NTo(int a[][KTMT], int m){  
    cout << "\nCac phan tu la so nguyen to (neu co)  
la:";  
    //sinh vien hoan thien ma lenh  
}  
int tinh_TongChan(int a[][KTMT], int m){  
    //sinh vien hoan thien ma lenh  
    //tinh tong cac phan tu la so chan  
    return 1;  
}  
void hienThi_Menu(){  
    cout << "\n\nChon 1 trong 6 so:";  
    cout << "\n1.  Nhap ma tran.";  
    cout << "\n2.  Xuat ma tran.";  
    cout << "\n3.  Xuat cac phan tu tren duong cheo  
phu.";  
    cout << "\n4.  Xuat tong cac phan tu la so  
chan.";  
    cout << "\n5.  Xuat cac phan tu la so nguyen  
to.";  
    cout << "\n0.  Thoat khoi chuong trinh.";  
}
```

```
void chon_Menu(int a[][KTMT], int &m){
    hienthi_Menu();
    int chon;
    do{
        cout << "\nNhap 1 so de chon chuc
nang...";
        cin >> chon;
        switch (chon){
            case 1:
                nhap_MaTran(a, m);
                hienThi_Menu();
                break;
            case 2:
                system("cls");
                xuat_MaTran(a, m);
                hienThi_Menu();
                break;
            case 3:
                system("cls");
                xuat_MaTran(a, m);
                xuat_Cheo_Phu(a, m);
                hienThi_Menu();
                break;
            case 4:
                system("cls");
                xuat_MaTran(a, m);
                cout << "\nTong cac phan tu la so
chan la:";
                // cout<<tong_chan(a,m);
                hienthi_Menu();
                break;
            case 5:
                system("cls");
                xuat_MaTran(a, m);
                //xuat_NTo(a,m);
                hienThi_Menu();
                break;
            default:
```

```
        break;  
    }  
} while (chon != 0);  
}
```