

**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG CAO ĐẲNG LÝ TỰ TRỌNG THÀNH PHỐ HỒ CHÍ MINH  
KHOA CÔNG NGHỆ THÔNG TIN**

★★★



**GIÁO TRÌNH**

**MÔN HỌC/MÔ ĐUN: THỰC HÀNH KỸ THUẬT LẬP TRÌNH  
NGÀNH/NGHỀ: CHUYÊN NGÀNH KHOA CÔNG NGHỆ THÔNG TIN  
TRÌNH ĐỘ: CAO ĐẲNG – TRUNG CẤP**

*Ban hành kèm theo Quyết định số: 1676/QĐ-LTT-ĐT ngày 31 tháng 12 năm 2020  
của Hiệu trưởng Trường Cao đẳng Lý Tự Trọng Thành phố Hồ Chí Minh*

TP. HCM, Năm 2023

## **TUYÊN BỐ BẢN QUYỀN**

Tài liệu này thuộc loại sách giáo trình lưu hành nội bộ nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

## LỜI GIỚI THIỆU

Sự phát triển công nghệ thông tin trong những năm vừa qua đã làm thay đổi bộ mặt kinh tế xã hội toàn cầu, trong đó việc tạo ra các phần mềm ứng dụng hỗ trợ công việc hàng ngày cũng như phục vụ giải trí cho con người ngày càng phổ biến. Để có thể tạo ra được các phần mềm ứng dụng yêu cầu phải có đội ngũ lập trình có năng lực chuyên môn tốt. Chính vì lý do đó, việc nghiên cứu, tìm hiểu về kỹ thuật lập trình trong máy tính cùng với sử dụng ngôn ngữ lập trình là việc rất cần thiết cho sinh viên chuyên ngành Công nghệ thông tin.

Tài liệu giảng dạy “Thực hành Kỹ thuật lập trình” được xây dựng với mục đích cung cấp cho sinh viên những kiến thức, kỹ năng cơ bản nhất, có tính hệ thống liên quan tới lập trình. Gồm phương pháp tư duy logic, các bước giải một bài toán lập trình Hướng dẫn sinh viên lập trình cơ bản bằng ngôn ngữ lập trình C/ C++. Nội dung giáo trình gồm 12 chương:

**Chương 1:** Khái quát về ngôn ngữ lập trình C/C++. Các kiểu dữ liệu cơ sở. Biểu thức – Toán tử - Câu lệnh . Nhập xuất dữ liệu

**Chương 2:** Cấu trúc rẽ nhánh

**Chương 3:** Vòng lặp for

**Chương 4:** Vòng lặp while, do...while

**Chương 5:** Hàm trong C/C++

**Chương 6:** Truyền tham số - hàm đệ quy

**Chương 7:** Mảng 1 chiều

**Chương 8:** Mảng 2 chiều

**Chương 9:** Chuỗi

**Chương 10:** Con trỏ

**Chương 11:** Kiểu cấu trúc

**Chương 12:** Ôn tập

Trong quá trình biên soạn giáo trình tác giả đã nhận được nhiều hỗ trợ từ các đồng nghiệp trong tổ bộ môn Khoa học máy tính nói riêng và khoa Công nghệ thông tin nói chung. Đây là giáo trình được biên soạn phục vụ giảng dạy môn **Thực hành Kỹ thuật lập trình**, cũng là sách dùng để tham khảo cho cá nhân và các đồng nghiệp yêu thích về ngôn ngữ lập trình C/C++. Trong quá trình biên soạn cũng không thể tránh khỏi những sai sót nhất định. Tác giả rất mong nhận được ý kiến đóng góp của các bạn đọc. Các ý kiến đóng góp xin gửi về:

Nguyễn Văn Minh - Khoa Công nghệ thông tin

Trường Cao đẳng Lý Tự Trọng Thành Phố Hồ Chí Minh

ĐT: 09888.77.405

Thành phố, Hồ Chí Minh, ngày tháng năm 2019

## MỤC LỤC

|                                                                                                                                      |    |
|--------------------------------------------------------------------------------------------------------------------------------------|----|
| CHƯƠNG 1: KHÁI QUÁT VỀ NGÔN NGỮ LẬP TRÌNH C/C++, CÁC KIỂU DỮ LIỆU CƠ SỞ.<br>BIỂU THỨC – TOÁN TỬ - CÂU LỆNH . NHẬP XUẤT DỮ LIỆU ..... | 6  |
| 1.1. Tóm tắt lý thuyết .....                                                                                                         | 6  |
| 1.2. Bài tập mẫu hướng dẫn thực hành.....                                                                                            | 10 |
| 1.3. Bài tập tại lớp:.....                                                                                                           | 14 |
| 1.4. Bài tập về nhà.....                                                                                                             | 15 |
| CHƯƠNG 2: CẤU TRÚC RỄ NHÁNH.....                                                                                                     | 16 |
| 2.1. Tóm tắt lý thuyết .....                                                                                                         | 16 |
| 2.2. Bài tập mẫu hướng dẫn thực hành.....                                                                                            | 21 |
| 2.3. Bài tập ở lớp .....                                                                                                             | 25 |
| 2.3. Bài tập về nhà: .....                                                                                                           | 26 |
| CHƯƠNG 3: VÒNG LẶP FOR .....                                                                                                         | 28 |
| 3.1. Tóm tắt lý thuyết .....                                                                                                         | 28 |
| 3.2. Bài tập mẫu hướng dẫn thực hành .....                                                                                           | 28 |
| 3.3. Bài tập ở lớp .....                                                                                                             | 31 |
| 3.4. Bài tập về nhà:.....                                                                                                            | 31 |
| CHƯƠNG 4: VÒNG LẶP WHILE , DO... WHILE.....                                                                                          | 33 |
| 4.1. Tóm tắt lý thuyết .....                                                                                                         | 33 |
| 4.2. Bài tập mẫu hướng dẫn thực hành .....                                                                                           | 34 |
| 4.3. Bài tập ở lớp .....                                                                                                             | 37 |
| 4.4. Bài tập về nhà:.....                                                                                                            | 38 |
| CHƯƠNG 5: HÀM TRONG C/C++ .....                                                                                                      | 40 |
| 5.1. Tóm tắt lý thuyết .....                                                                                                         | 40 |
| 5.2. Bài tập mẫu hướng dẫn thực hành .....                                                                                           | 42 |
| 5.3. Bài tập ở lớp .....                                                                                                             | 44 |
| 5.4. Bài tập về nhà:.....                                                                                                            | 45 |
| CHƯƠNG 6: TRUYỀN THAM SỐ - HÀM ĐỆ QUY.....                                                                                           | 47 |
| 6.1. Tóm tắt lý thuyết .....                                                                                                         | 47 |
| 6.2. Bài tập mẫu hướng dẫn thực hành .....                                                                                           | 48 |
| 6.3. Bài tập ở lớp .....                                                                                                             | 50 |
| 6.4. Bài tập về nhà:.....                                                                                                            | 51 |
| CHƯƠNG 7: MẢNG MỘT CHIỀU .....                                                                                                       | 51 |
| 7.1. Tóm tắt lý thuyết .....                                                                                                         | 52 |
| 7.2. Bài tập mẫu hướng dẫn thực hành .....                                                                                           | 54 |
| 7.3. Bài tập ở lớp .....                                                                                                             | 56 |
| 7.4. Bài tập về nhà:.....                                                                                                            | 57 |

|                                               |    |
|-----------------------------------------------|----|
| CHƯƠNG 8: MẢNG HAI CHIỀU.....                 | 60 |
| 8.1. Tóm tắt lý thuyết .....                  | 60 |
| 8.2. Bài tập mẫu hướng dẫn thực hành. ....    | 61 |
| 8.3. Bài tập ở lớp .....                      | 66 |
| 8.4. Bài tập về nhà:.....                     | 66 |
| CHƯƠNG 9: CHUỖI .....                         | 68 |
| 9.1. Tóm tắt lý thuyết .....                  | 68 |
| 9.2. Bài tập mẫu hướng dẫn thực hành. ....    | 69 |
| 9.3. Bài tập ở lớp .....                      | 70 |
| 9.4. Bài tập về nhà:.....                     | 71 |
| CHƯƠNG 10: CON TRỎ.....                       | 71 |
| 10.1. Tóm tắt lý thuyết .....                 | 72 |
| 10.2. Bài tập mẫu hướng dẫn thực hành. ....   | 73 |
| 10.3. Bài tập ở lớp .....                     | 75 |
| 10.4. Bài tập về nhà:.....                    | 75 |
| CHƯƠNG 11: KIỂU CẤU TRÚC.....                 | 77 |
| 11.1. Tóm tắt lý thuyết .....                 | 77 |
| 11.2. Bài tập mẫu hướng dẫn thực hành. ....   | 79 |
| 11.3. Bài tập ở lớp .....                     | 81 |
| 11.4. Bài tập về nhà:.....                    | 82 |
| CHƯƠNG 12: Vận dụng vào bài toán cụ thể ..... | 84 |
| 1.1. Tóm tắt lý thuyết .....                  | 84 |
| 1. Hàm đệ quy:.....                           | 84 |
| 2. Chuỗi.....                                 | 84 |
| 3. Mảng 1 chiều.....                          | 84 |
| 4. Mảng cấu trúc .....                        | 84 |
| 12.2. Bài tập mẫu hướng dẫn thực hành. ....   | 84 |
| 12.3. Bài tập ở lớp .....                     | 86 |
| 12.4. Bài tập về nhà:.....                    | 87 |

## **GIÁO TRÌNH MÔN HỌC/MÔ ĐUN**

**Tên môn học/mô đun: THỰC HÀNH KỸ THUẬT LẬP TRÌNH**

**Mã môn học/mô đun: TTC1414**

**Vị trí, tính chất, ý nghĩa và vai trò của môn học/mô đun:**

- **Vị trí:** Học song song với môn học Kỹ thuật lập trình.
- **Tính chất:** Là môn học bắt buộc nằm trong các môn học cơ sở. Môn học này yêu cầu phải có tư duy logic và kiến thức về toán.
- **Ý nghĩa và vai trò của môn học/mô đun:** Kỹ thuật lập trình là môn học lý thuyết cơ sở của ngành thuộc khoa Công nghệ thông tin. Người học được trang bị kiến thức về các khái niệm cơ bản về ngôn ngữ lập trình, phân tích các yêu cầu bài toán để viết chương trình cho máy tính thực hiện và cho ra kết quả. Rèn luyện kỹ năng lập trình cho sinh viên

**Mục tiêu của môn học/mô đun:**

- **Về kiến thức:** Sau khi học xong sinh viên có các kiến thức cơ bản về kỹ thuật lập trình như hàm, hàm đệ quy, mảng, chuỗi, con trỏ, struct.
- **Về kỹ năng:**
  - + Sử dụng thành thạo cấu trúc rẽ nhánh, vòng lặp, hàm, hàm đệ quy, mảng, chuỗi, con trỏ, struct, file trong lập trình bằng ngôn ngữ C/C++
  - + Lập trình được các bài toán thực tế dùng C/C++
  - + Giải quyết được các bài toán thực tế dùng C/C++
- **Về năng lực tự chủ và trách nhiệm:**
  - + Có khả năng tự duy độc lập trong lập trình.
  - + Có khả năng tự tìm tòi, học hỏi, tích lũy kinh nghiệm liên quan đến lập trình.

**Nội dung của môn học/mô đun:**

## CHƯƠNG 1: KHÁI QUÁT VỀ NGÔN NGỮ LẬP TRÌNH C/C++. CÁC KIỂU DỮ LIỆU CƠ SỞ. BIỂU THỨC – TOÁN TỬ - CÂU LỆNH. NHẬP XUẤT DỮ LIỆU

**Giới thiệu:** *Chương 1* – Các nội dung được trình bày về môi trường lập trình, khai báo biến, tạo project lưu trữ, nhập xuất dữ liệu.

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Đặt được tên cho các biến theo đúng yêu cầu của ngôn ngữ lập trình.
- Tạo được chú thích và đoạn code cơ bản
- Tạo và lưu project bằng môi trường studio .NET
- Nhập xuất được dữ liệu
- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

**Nội dung chính:**

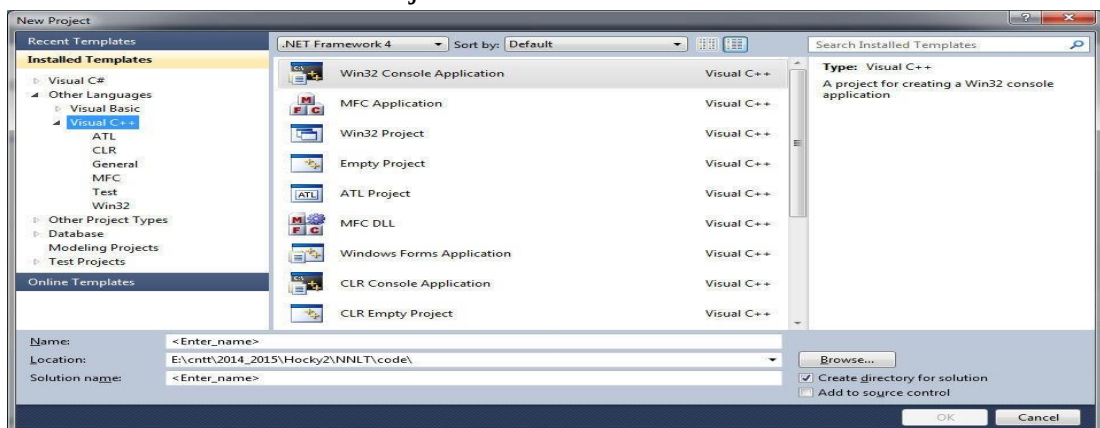
### 1.1. Tóm tắt lý thuyết

#### 1.1.1. Sử dụng Visual C++.Net để viết chương trình

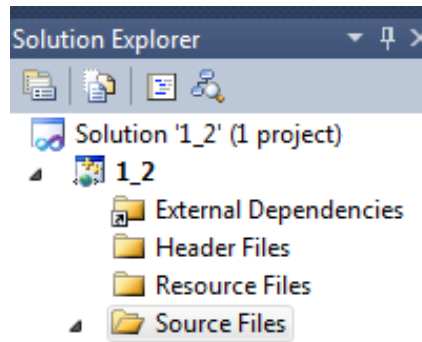
- Khởi động Visual Studio.Net



- Vào File → New → Project



Chọn Visual C++, Empty Project, Đặt tên (Name) và chọn thư mục lưu Project (Location) , Chọn OK.



- Nhấn chuột phải vào Source files trên thanh Solution Explorer (bên phải) → Add → New Item
- Chọn C++ File → Đặt tên File (Name) → Add.
  - Sau đó viết code vào màn hình soạn thảo
  - Thực thi chương trình: Nhấn phím F5 hoặc vào Debug → Start Debugging.

#### 1.1.2. Các kiểu dữ liệu cơ sở

- Kiểu số nguyên

| Kiểu (Type) | Độ lớn (Byte) | Miền giá trị (Range)              |
|-------------|---------------|-----------------------------------|
| char        | 1             | -128 ... +127                     |
| int         | 2             | -32.768 ... +32.767               |
| short       | 2             | -32.768 ... +32.767               |
| long        | 4             | -2.147.483.648 ... +2.147.483.647 |

- Kiểu số thực:

| Kiểu (Type) | Độ lớn (Byte) | Miền giá trị (Range)                           |
|-------------|---------------|------------------------------------------------|
| float (*)   | 4             | $3.4 \cdot 10^{-38}$ ... $3.4 \cdot 10^{38}$   |
| double (**) | 8             | $1.7 \cdot 10^{-308}$ ... $1.7 \cdot 10^{308}$ |

- Kiểu ký tự:

- Tên kiểu: char
- Miền giá trị: 256 ký tự trong bảng mã ASCII.

Kiểu ký tự chính là kiểu số nguyên và nó lưu tất cả dữ liệu ở dạng số và không lưu trực tiếp ký tự mà chỉ lưu mã ASCII của ký tự đó.

Mã ASCII từ 'A' đến 'Z' là

65 đến 90 Mã ASCII từ 'a'

đến 'z' là 97 đến 122.

- Kiểu luận lý: Trong C chuẩn, không có kiểu luận lý. Giá trị sai (false) là 0, giá trị đúng (true) là khác 0 (thường là 1). Trong C++, kiểu luận lý là bool

➤ Biến – hằng – câu lệnh và biểu thức

- Biến

- Cú pháp:

```
<kiểu> <tên biến>;
```

```
<kiểu> <tên biến 1>, <tên biến 2>;
```

```
Float a;
```

```
int j, k;
```

- Hằng

- Cú pháp:

```
#define <tên hằng> <giá trị>
```

Hoặc sử dụng từ khóa const.

- Ví dụ:

```
#define MAX 100 // Không có dấu;
```

```
#define PI 3.14 // Không
```

```
có dấu ; const int MAX = 100;
```

```
const float PI = 3.14;
```

- Câu lệnh: được kết thúc bằng dấu chấm phẩy (;)

Câu lệnh đơn: chỉ gồm 1 câu lệnh

Câu lệnh phức: gồm nhiều câu lệnh đơn (từ 2 câu lệnh đơn trở lên) được bao bởi { và }

Ví dụ:

```
a = 2912; // Câu lệnh đơn
```

```
{ // Câu lệnh phức/khối lệnh
```

```
    a = 2912;
```

```
    b = 1706;
```

```
}
```

- Biểu thức: được tạo thành từ các toán tử (Operator) và các toán hạng (Operand).

Ví dụ:

```
int a = 2 + 3 * 4;
```

➤ Chuỗi định dạng

| Đặc tả               | Ý nghĩa             |                         |
|----------------------|---------------------|-------------------------|
| <code>%c</code>      | Ký tự               | char                    |
| <code>%d, %ld</code> | Số nguyên có dấu    | char, int, short, long  |
| <code>%f, %lf</code> | Số thực             | float, double           |
| <code>%s</code>      | Chuỗi ký tự         | char[], char*           |
| <code>%u</code>      | Số nguyên không dấu | unsigned int/short/long |

| Ký tự điều khiển | Ý nghĩa          |
|------------------|------------------|
| <code>\a</code>  | Tiếng chuông     |
| <code>\b</code>  | Lùi lại một bước |
| <code>\n</code>  | Xuống dòng       |
| <code>\t</code>  | Dấu tab          |
| <code>\\</code>  | In dấu \         |
| <code>\?</code>  | In dấu ?         |
| <code>\"</code>  | In dấu "         |

### ➤ Ép kiểu dữ liệu

- Ngầm định: Kiểu dữ liệu đích phải có dãy giá trị lớn hơn kiểu dữ liệu nguồn Ví dụ: chuyển từ int sang float.

```
int a = 10;
```

```
float f = a;
```

- Tường minh: Thay đổi kiểu dữ liệu có dãy giá trị lớn sang dãy giá trị nhỏ hơn Ví dụ: chuyển từ float sang int

```
float f = 10.51;
```

```
int a = (int) f;
```

### 1.1.3. Câu lệnh nhập xuất

- Thư viện:

```
#include <stdio.h> (standard input/output)
```

- Câu lệnh nhập

```
scanf ("<chuỗi định dạng>", <các đối số>)
```

Ví dụ:

```
scanf ("%d", &a); // Nhập giá trị cho biến a
```

- Câu lệnh xuất

```
printf ("<chuỗi cần xuất>/[<định dạng>]", [<các đối số>])
```

Ví dụ:

```
printf ("Hello World");
```

```
printf ("Gia tri cua bien c la: %d", c);
```

### 1. Cách sử dụng kiểu char

- Miền giá trị: 256 ký tự trong bảng mã ASCII.

- Chính là kiểu số nguyên do:
  - Lưu tất cả dữ liệu ở dạng số.
  - Không lưu trực tiếp ký tự mà chỉ lưu mã ASCII của ký tự đó.
- Sử dụng char như là số nguyên hoặc ký tự đều được Ví dụ:

```
char c = 'A'
printf("%d", c) → in ra 65
printf("%c", c) → in ra 'A'
```

## 2. Sử dụng toán tử 1 ngôi

- Giống nhau giữa ++<biến> và <biến>++
  - Đều tăng giá trị của <biến> thêm 1 đơn vị
  - Cách sử dụng như nhau nếu chúng nằm độc lập
- Khác nhau giữa ++<biến> và <biến>++
  - Khi chúng nằm trong một biểu thức.
  - ++<biến> sẽ tăng <biến> trước khi thực hiện biểu thức
  - <biến>++ sẽ thực hiện biểu thức trước khi tăng <biến> Ví dụ:

```
int i = 1;
int b = 10 + ++i; // (1) sẽ khác với int b = 10 + i++(2).
```

Biểu thức (1) b sẽ có kết quả là 12. Biểu thức (2) b sẽ có kết quả là 11.

- Tương tự cho toán tử --

### 1.1.4. Toán tử điều kiện

- Cú pháp: <biểu thức 1> ? <biểu thức 2> : <biểu thức 3>  
<biểu thức 1> đúng thì giá trị là <biểu thức 2>.  
<biểu thức 1> sai thì giá trị là <biểu thức 3>.

Ví dụ:

```
s1 = (1 > 2) ? 2912 : 1706;
int s2 = 0;
1 < 2 ? s2 = 2912 : s2 = 1706;
```

Kết quả: s1 = 1706, s2 = 2912

### 1.1.5 Các bước thực hiện bài thực hành

- Bước 1: Phân tích bài toán.
- Bước 2: Cài đặt chương trình.
- Bước 3: Kiểm tra và chạy thử.

## 1.2. Bài tập mẫu hướng dẫn thực hành

### 1. Viết chương trình xuất ra đoạn thông báo:

*"Chao ban!"*

*Đây là chương trình C đầu tiên.*

*Vui lòng nhấn phím Enter để kết thúc."*

Tạo project mới như phần I.1, viết chương trình sau trong màn hình soạn thảo:

- *Input: không có*
- *Output : đoạn thông báo*

```
#include<stdio.h>
#include<conio.h>
void main()
{
    printf("Chao ban!\n");
    printf("Day la chuong trinh C dau tien.\n ");
    printf("\t Vui long nhan phim Enter de ket
thuc.");
    getch();
}
```

2. Viết chương trình nhập vào 1 số nguyên, xuất ra màn hình số vừa nhập.

- *Input: 1 số nguyên a bất kỳ*
- *Output : số nguyên a*

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a;
    printf("Nhap so nguyen a:
"); scanf("%d", &a);
    printf("So nguyen vua nhap la: %d", a);
    getch();
}
```

3. Viết chương trình nhập vào 2 số nguyên, tính và xuất kết quả tổng, tích, hiệu và thương 2 số nguyên vừa nhập.

- *Input: 2 số nguyên a, b bất kỳ*
- *Output : các kết quả tổng, hiệu, tích, thương của a và b*

```
#include<stdio.h>
#include<conio.h>

void main()
{
    int a, b;
    printf("Nhap 2 so nguyen:
"); scanf("%d%d", &a, &b);

    int tong = a + b;
    int hieu = a - b;
    int tich = a * b;

    //Ep kieu tu so nguyen sang so thuc
    float thuong = (float)a/(float)b;
    printf("Tong = %d\n", tong);
    printf("Hieu = %d\n", hieu);
    printf("Tich = %d\n", tich);

    //In ra so thuc co 1 chu so thap phan
    printf ("Thuong = %.1f", thuong);
    getch();
}
```

#### 4. Tính tiền khách ở khách sạn trong tháng.

- Nhập vào ngày đến ở khách sạn, nhập ngày rời khỏi khách sạn (giả sử ngày đến và ngày đi là số nguyên dương trong tháng (từ ngày 1 tới 31).
- Tính tổng số ngày khách đã ở trong tháng.
- Tính tiền khách phải trả, biết rằng đơn giá tuần là 650 và đơn giá ngày là 100.

- Input: giá trị ngày đến và ngày đi (1 → 31); đơn giá tuần và ngày
- Output : số ngày ở và tổng tiền phải trả
- Hướng dẫn:
  - + Nhập ngày đến và ngày đi là số nguyên.
  - + Tính số ngày ở → tính số tuần và ngày lẻ
  - + Tính thành tiền theo đơn giá tuần và ngày lẻ
  - + Xuất kết quả

5. Nhập chương trình sau, thực thi và xem kết quả để phân biệt sự giống và khác nhau của toán tử 1 ngôi.

```
#include<stdio.h>

#include<conio.h>

void main()
{
    int i = 1;
    printf("i = %d\n", i); int bt = 10 + ++i;
    printf("Ket qua cua bieu thuc: 10 + ++i la: %d; gia
tri cua i la: %d\n", bt, i);

    i = 1;
    printf("Khoi tao lai gia tri i = %d\n", i);
    bt = 10 + i++;
    printf("Ket qua cua bieu thuc: 10 + i++ la: %d; gia
tri cua i la: %d\n", bt, i);
    getch();
}
```

6. Nhập mã ASCII. Nếu mã ASCII là của 26 chữ cái tiếng Anh (A -> Z) thì in ra kết quả là “là chữ cái”, ngược lại in ra ký tự tương ứng với mã đó.

```
#include<stdio.h>

#include<conio.h>

void
main()
{
```

```
int as;

printf("Nhap ma ASCII: "); scanf("%d", &as);

((as >= 'A' && as <= 'Z') || (as >= 'a' && as <=
'z'))?(printf("La chu cai")):(printf("Ky tu: %c", as));
```

### 1.3. Bài tập tại lớp:

1. Viết chương trình xuất ra (đúng định dạng) một đoạn trong bài thơ

Lượm: Chú bé loắt choắt

Cái sắc xinh xinh

Cái chân thoăn thoắt

Cái đầu nghênh nghênh.

Hướng dẫn: sử dụng các kí tự định dạng trang 4

2. Viết chương trình nhập vào số thực. In ra phần nguyên của số thực đó.

Hướng dẫn:

1 số thực được nhập từ bàn phím

sử dụng phép toán ép kiểu để tính phần nguyên

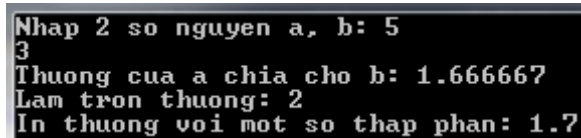
3. Viết chương trình nhập bán kính và tính chu vi, diện tích hình tròn.

Gợi ý: khai báo hằng số pi để thực hiện.

4. Nhập vào 2 số nguyên a,b. Tìm số lớn nhất trong 2 số.

Gợi ý: dùng phép toán điều kiện

5. Viết chương trình để có kết quả như sau:



```
Nhap 2 so nguyen a, b: 5
3
Thuong cua a chia cho b: 1.666667
Lam tron thuong: 2
In thuong voi mot so thap phan: 1.7
```

Hướng dẫn:

2 số nguyên được nhập từ bàn phím

Tính thương của 2 số nguyên (bài tập mẫu hướng dẫn 6 trang 6)

Tính làm tròn thương bằng cách +0.5 cho thương và ép về số nguyên

Xuất phần thương có 1 chữ số thập phân

6. Viết chương trình để có kết quả như sau:

```
Tính tiền thuê phòng
Đơn giá tuần: 500000. Đơn giá ngày: 100000
Nhập vào tổng số ngày thuê phòng: 9
Số tuần mà khách thuê: 1, số ngày khách thuê: 2
Tiền phải trả: 700000
```

#### Hướng dẫn:

2 số nguyên (đơn giá tuần, đơn giá ngày) là hằng số

1 số nguyên (tổng số ngày) được nhập từ phím

Tính số tuần dựa vào phần nguyên khi chia tổng số ngày cho 7

Tính số ngày lẻ dựa vào hiệu của tổng ngày và (số tuần \* 7 hoặc dùng phép %)

Tính tiền phải trả dựa vào tổng tiền thuê theo tuần và tiền thuê theo ngày lẻ.

7. Viết chương trình để tính tiền sân bóng đá như sau:

Tính tiền giờ dựa vào giờ vào và giờ ra. Mỗi giờ là 200,000

Tính tiền nước uống dựa vào số bình nước đã uống. Mỗi bình nước 20,000

Tính tổng tiền: tiền giờ + tiền nước uống

### **1.4. Bài tập về nhà**

1. Viết chương trình hoán đổi giá trị của 2 số nguyên được nhập.

2. Viết chương trình nhập vào chiều dài và chiều rộng của một hình chữ nhật, tính và in ra màn hình:

a. Chu vi hình chữ nhật

b. Diện tích hình chữ nhật

3. Nhập điểm thi và hệ số 3 môn Toán, Lý, Hóa của một sinh viên. Tính điểm trung bình của sinh viên đó.

4. Nhập 2 cạnh của hình chữ nhật (là số nguyên). Tính diện tích, chu vi và độ dài đường chéo (số thực) của hình chữ nhật.

5. Nhập vào 1 ký tự, kiểm tra ký tự đó là: kí tự in hoa, kí tự thường hay ký tự khác.

6. Nhập 1 ký tự. Kiểm tra ký tự đó có thuộc 26 chữ cái tiếng Anh hay không?

7. Viết chương trình giải phương trình bậc nhất:  $ax + b = 0$  ( $a$  khác 0), với  $a, b$  là số nguyên,  $x$  là số thực.

## CHƯƠNG 2: CẤU TRÚC RỄ NHÁNH

**Giới thiệu:** *Chương 1* – Các nội dung được trình bày về cấu trúc if, if...else, và if...else lồng nhau

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Vận dụng thành thạo cấu trúc điều kiện if .. else
- Vận dụng thành thạo cấu trúc switch .. case.
- Viết và chạy chương trình cho ra kết quả chính xác.
- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

**Nội dung chính:**

### 2.1. Tóm tắt lý thuyết

Rẽ nhánh: là tại một thời điểm sẽ có nhiều lựa chọn có thể xảy ra nhưng chỉ xảy ra một trường hợp duy nhất.

#### 2.1.1. Câu lệnh if:

Sử dụng khi rẽ nhánh chỉ có một trường hợp duy nhất. Nếu thỏa điều kiện thì thực hiện, còn không thỏa thì không thực hiện (không làm gì hết).

**Ví dụ:** Sinh viên có điểm trung bình lớn hơn 8.0 được nhận học bổng. Sinh viên khác: không có yêu cầu gì cả.

Cú pháp:

if(<biểu thức điều kiện>)

{

    //Câu lệnh xử lý khi thỏa điều kiện

}

Ví dụ:

if(dtb>8.0)

{

    printf("Ban nhan duoc hoc bong");

}

#### 2.1.2. Câu lệnh if...else...

- Sử dụng khi rẽ nhánh chỉ có 2 trường hợp trái ngược nhau xảy ra. Nếu không thực hiện trường hợp này thì phải thực hiện trường hợp kia và ngược lại.

- Ví dụ: Nếu sinh viên có điểm trung bình lớn hơn hoặc bằng 4.0 thì đậu, ngược lại thì rớt. Ta thấy trong ví dụ này, bất kỳ sinh viên nào cũng chỉ có thể rơi vào trường hợp “đậu” hoặc “rớt”, không có trường hợp khác.
- Sinh viên có điểm trung bình lớn hơn hoặc bằng 4.0: “đậu”. Sinh viên lớp khác: “rớt”.
- Cú pháp:  
if(<biểu thức điều kiện>)

```
{  
    // Câu lệnh khi thỏa điều kiện  
}  
else  
{  
    //Câu lệnh khi không thỏa điều kiện  
}
```

Ví dụ:

```
if(dtb>=4.0)  
    printf("Đau");  
else  
    printf("rot");
```

### 2.1.3. Câu if lồng nhau:

- Sử dụng khi rẽ nhánh có từ 3 trường hợp trở lên.
- Cú pháp

```
if(<biểu thức điều kiện 1>)  
{  
    // Câu lệnh khi thỏa điều kiện 1  
}  
else
```

```
if(<biểu thức điều kiện 2>
{
    //Câu lệnh khi không thỏa điều kiện 1 nhưng thỏa điều kiện 2
}
else
    if(<biểu thức điều kiện 3>)
    {
        //Câu lệnh khi không thỏa điều kiện 2 nhưng thỏa điều
        kiện 3
    }
    else
    {
        //Câu lệnh khi không thỏa các điều kiện trên
    }
```

- Ví dụ: Căn cứ vào điểm trung bình ta xếp loại sinh viên như sau:
  - + Nếu  $đtb \geq 8.0$  thì xếp loại giỏi
  - + Nếu  $8.0 > đtb \geq 6.5$  thì xếp loại khá
  - + Nếu  $6.5 > đtb \geq 5.0$  thì xếp loại trung bình
  - + Còn lại thì xếp loại yếu

Như vậy, ta viết đoạn chương trình thực hiện yêu cầu trên bằng cách sử dụng cấu trúc if lồng nhau như sau:

```
if(đtb >= 8.0)
    printf("Xep loại gioi");
else
    if(đtb >= 6.5)
        printf("Xep loại kha");
    else
        if(đtb >= 5.0)
            printf("Xep loại trung binh");
        else
```

```
printf("Xep loại Yeu");
```

### Một vài chú ý:

- Biểu thức điều kiện trong cấu trúc if, if...else... phải là một mệnh đề có giá trị đúng hoặc sai.
- Biểu thức điều kiện có thể chỉ gồm một biểu thức đơn giản hoặc có thể là sự kết hợp của nhiều biểu thức điều kiện thông qua các phép toán và (&&), hoặc (||), phủ định (!).

#### 2.1.4. Câu lệnh switch...case...

- Sử dụng khi bài toán rẽ nhánh có nhiều trường hợp (thường là từ 3 trở lên) và mỗi trường hợp là một giá trị cụ thể (có kiểu giá trị số nguyên hoặc ký tự)
- VD: Nếu xếp hạng 1 bạn được thưởng 10 cuốn tập, xếp hạng 2 bạn được thưởng 8 cuốn tập, xếp hạng 3 bạn được thưởng 5 cuốn tập, xếp hạng 4 bạn được thưởng 3 cuốn tập, còn lại bạn không được thưởng.

Hạng 1: 10 cuốn tập

Hạng 2: 8 cuốn tập

Hạng 3: 5 cuốn tập

Hạng 4: 3 cuốn tập

Còn lại: 0 cuốn tập

Cú pháp:

```
switch(<bieuthuc>)
```

```
{
```

```
    case <giá trị 1>:
```

```
        // Xử lý khi giá trị biểu thức bằng 1
```

```
        break;
```

```
    case <giá trị 2>:
```

```
        // Xử lý khi giá trị biểu thức bằng 2
```

```
        break;
```

```
    case <giá trị 3>:
```

```
        // Xử lý khi giá trị biểu thức bằng 3
```

```
        break;
```

```
    ...
```

```
    default:
```

```
// Xử lý khi giá trị biểu thức không bằng các giá trị trên
```

```
}
```

Ví dụ:

```
switch(xephang)
```

```
{
```

```
    case 1:
```

```
        printf("Thuong 10 cuon tap");
```

```
        break;
```

```
    case 2:
```

```
        printf("Thuong 8 cuon tap");
```

```
        break;
```

```
    case 3:
```

```
        printf("Thuong 5 cuon tap");
```

```
        break;
```

```
    case 4:
```

```
        printf("Thuong 3 cuon tap");
```

```
        break;
```

```
    default:
```

```
        printf(" Ban khong duoc thuong tap ");
```

```
}
```

Mỗi trường hợp (case) trong câu lệnh switch phải kết thúc bằng lệnh break;

Kiểu trả về của <biểu thức> trong câu lệnh switch phải có cùng kiểu với các <gia tri1>, <gia tri 2>,... Cần xác định đúng biểu thức rẽ nhánh trong câu lệnh switch...case. Biểu thức rẽ nhánh là biểu thức có thể nhận các giá trị khác nhau nhưng trong mỗi trường hợp chỉ nhận 1 giá trị duy nhất. Biểu thức có thể là một biến.

## 2.2. Bài tập mẫu hướng dẫn thực hành

1. Nhập vào điểm trung bình môn Nhập môn lập trình của sinh viên. In ra màn hình cho biết sinh viên đó đậu hay rớt biết rằng nếu điểm trung bình lớn hơn hoặc bằng 4.0 thì đậu ngược lại là rớt.

### **Bước 1: Phân tích bài toán**

- Nhập điểm trung bình
- Xuất ra kết quả đậu hay rớt dựa vào điểm trung bình đã nhập.
- 01 biến dtb (điểm trung bình) có kiểu là số thực
- Bài toán có 2 trường hợp là đậu, rớt => sử dụng cấu trúc if...else...
- Biểu thức điều kiện:  
dtb >= 4.0

### **Bước 2: Cài đặt chương trình**

- Tạo project mới (giống các bài thực hành trước): File ☐ new ☐ Project
- Chọn ngôn ngữ là C++
- Khai báo thư viện và viết hàm main #include "stdio.h" #include "conio.h"
- Khai báo và nhập dữ liệu

```
float dtb;
```

```
printf("nhapdiemtrungbinhmonNNLT");  
scanf("%f",&dtb);
```

- Cài đặt câu lệnh rẽ nhánh

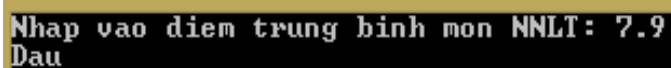
```
if(dtb >= 4.0)
```

```
    printf("Đau");
```

```
else
```

```
    printf("Rớt");
```

### **Bước 3: Kiểm tra, sửa lỗi và chạy thử**



```
Nhap vao diem trung binh mon NNLT: 7.9  
Đau
```

2. Nhập vào một số nguyên, cho biết số vừa nhập có là số chẵn hay không?

### **Bước 1: Phân tích bài toán**

- Nhập số nguyên bất kỳ
- Xuất ra kết quả số chẵn hoặc không làm gì cả
- 01 biến n có kiểu là số nguyên (int)
- Bài toán có 1 trường hợp là số chẵn => sử dụng cấu trúc if...
- Biểu thức điều kiện: n%2==0

## Bước 2: Cài đặt chương trình

- Khai báo thư viện và viết hàm main

```
#include "stdio.h"
```

```
#include "conio.h"
```

- Khai báo và nhập dữ liệu

```
int n;
```

```
printf("nhapvapoimsonguyen:");
```

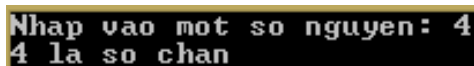
```
scanf("%d",&n);
```

- Cài đặt câu lệnh rẽ nhánh

```
if(n%2==0)
```

```
printf("%d la so chan",n);
```

## Bước 3: Kiểm tra, sửa lỗi và chạy thử



```
Nhap vao mot so nguyen: 4
4 la so chan
```

3. Nhập vào một năm, cho biết năm vừa nhập có phải là năm nhuận hay không. Biết rằng *năm nhuận là năm chia hết cho 4 mà không chia hết cho 100 hoặc năm chia hết cho 400?* Ví dụ: 1996 là năm nhuận vì 1996 chia hết cho 4 và không chia hết cho 100; 2100 không là năm nhuận vì chia hết cho 4 và 100 nhưng không chia hết cho 400.

## Bước 1: Phân tích bài toán

- Nhập vào một năm (số nguyên)
- Xuất ra kết quả là năm nhuận hoặc năm không nhuận
- 01 biến nam (năm) có kiểu là số nguyên (int)
- Bài toán có 2 trường hợp là năm nhuận, năm không nhuận => sử dụng cấu trúc if...else...
- Biểu thức điều kiện: năm chia hết cho 4 mà không chia hết cho 100 hoặc năm chia hết cho 400

## Bước 2: Cài đặt chương trình

- Khai báo thư viện và viết hàm main
- Khai báo và nhập dữ liệu
- Cài đặt câu lệnh rẽ nhánh

### **Bước 3: Kiểm tra, sửa lỗi và chạy thử**

```
Nhap vào nam ban muon kiem tra: 1996
1996 la nam nhuan
```

4. Một môn học lý thuyết thường có 03 cột điểm hệ số 10: điểm tiểu luận, điểm giữa kỳ, điểm cuối kỳ có tỉ lệ lần lượt là 20%, 30%, 50%. Hãy cho biết điểm chữ của môn học khi người dùng nhập vào điểm tiểu luận, điểm giữa kỳ và điểm cuối kỳ. Giả sử rằng cách quy đổi điểm hệ số 10 sang điểm chữ như sau:

| Điểm hệ 10                         | điểm chữ |
|------------------------------------|----------|
| Điểm hệ 10 $\geq 8.5$              | A        |
| $8.5 > \text{điểm hệ 10} \geq 7.0$ | B        |
| $7.0 > \text{điểm hệ 10} \geq 5.5$ | C        |
| $5.5 > \text{điểm hệ 10} \geq 4.0$ | D        |
| $4.0 > \text{điểm hệ 10}$          | F        |

### **Bước 1: Phân tích bài toán**

- Nhập vào điểm tiểu luận, giữa kỳ, cuối kỳ
- Xuất ra điểm chữ của môn học đó theo qui tắc đã cho
- 04 biến diemTL, diemGK, diemCK, diemTB là các số thực (float) và
- $\text{diemTB} = 0.2 * \text{diemTL} + 0.3 * \text{diemGK} + 0.5 * \text{diemCK}$
- Bài toán có 5 trường hợp điểm chữ  $\Rightarrow$  sử dụng cấu trúc if lồng nhau
- Biểu thức điều kiện:
  - + Điều kiện 1:  $\text{diemTB} \geq 8.5$
  - + Điều kiện 2:  $\text{diemTB} \geq 7.0$
  - + Điều kiện 3:  $\text{diemTB} \geq 5.5$
  - + Điều kiện 4:  $\text{diemTB} \geq 4.0$

### **Bước 2: Cài đặt chương trình**

- Khai báo thư viện và viết hàm main
- Khai báo, nhập dữ liệu và tính toán
- Cài đặt câu lệnh rẽ nhánh

### **Bước 3: Kiểm tra, sửa lỗi và chạy thử**

```
Nhap vào diem tieu luan, diem giua ky va diem cuoi ky: 7 9 6
Ban duoc diem B
```

5. Nhập vào 2 số nguyên a và b, nhập vào một phép toán +, -, \*, /. Hãy in ra màn

hình kết quả của phép toán vừa nhập với 2 số hạng là a, b đã nhập

### **Bước 1: Phân tích bài toán**

- Nhập 2 số nguyên bất kỳ, nhập 1 phép toán
- Xuất ra kết quả của phép toán vừa nhập
- 02 biến số nguyên (int), 01 biến lưu phép toán (biến pheptoan, kiểu char), 01 biến lưu kết quả (biến kết quả kiểu int)
- Bài toán có 4 trường hợp của phép toán: +, -, \*, / => dùng cấu trúc switch...case
- Biểu thức rẽ nhánh: biến pheptoan

### **Bước 2: Cài đặt chương trình**

- Khai báo và nhập dữ liệu  

```
int a,b, ketqua=0;
printf("nhap vao 2 so nguyen:");
scanf("%d%d",&a,&b);
char pheptoan;
printf("nhap vao phep toan ban muon tinh:");
scanf("%c", &pheptoan);
```

Lưu ý: trước khi nhập phép toán phải có câu lệnh **flushall()**; để xóa bỏ các giá trị còn lưu trong vùng đệm

- Cài đặt câu lệnh rẽ nhánh

```
switch(pheptoan)
{
    case '+': ketqua=a+b; break;
    case '-': ketqua=a-b; break;
    case '*': ketqua=a*b; break;
    case '/': ketqua=a/b; break;
}
```

### **Bước 3: Kiểm tra, sửa lỗi và chạy thử**

6. Nhập một tháng (nhập bằng số), in ra màn hình tháng đó bằng tiếng Anh tương ứng (January, February, March, April, May, June, July, August, September, October, November, December). Nếu nhập tháng không đúng in ra màn hình "Month is invalid"

### **Bước 1: Phân tích bài toán**

- Nhập vào một số tháng bằng số nguyên
- Xuất tháng đó ra màn hình bằng tiếng Anh tương ứng
- 01 biến thang có kiểu là số nguyên (int)

- Bài toán có 12 trường hợp hợp lệ và 01 trường hợp không hợp lệ => sử dụng switch...case...
- Biểu thức rẽ nhánh: sử dụng biến thang

## **Bước 2: Cài đặt chương trình**

- Khai báo thư viện và viết hàm main
- Khai báo và nhập dữ liệu
- Cài đặt câu lệnh rẽ nhánh: switch - case

## **Bước 3: Kiểm tra, sửa lỗi và chạy thử**

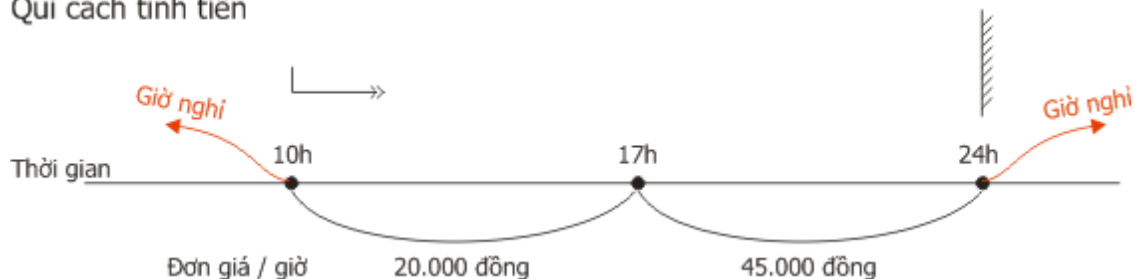
### **2.3. Bài tập ở lớp**

1. Viết chương trình nhập vào 3 số a, b. Tìm số lớn nhất giữa 3 số.
2. Viết chương trình nhập vào một số nguyên, kiểm tra số đã nhập là số âm hay số dương.
3. Viết chương trình nhập vào 2 số nguyên, so sánh 2 giá trị vừa nhập vào (“Bằng nhau, nhỏ hơn, lớn hơn”).
4. Nhập vào thứ tự của tháng, cho biết tháng đó có bao nhiêu ngày.
5. Viết một chương trình tính chu vi, diện tích theo dạng menu cho phép người dùng chọn lựa một trong các hình bên dưới (dùng cấu trúc switch...case)
  - Hình vuông
  - Hình chữ nhật
  - Hình tròn
6. Viết chương trình nhập vào 2 số a,b. Kiểm tra a có là bội số của b không.
7. Viết chương trình nhập vào đơn giá 1 mặt hàng, và số lượng bán của mặt hàng. Tính tiền khách phải trả, với thông tin như sau:
  - Thành tiền: đơn giá \* số lượng
  - Giảm giá: Nếu thành tiền > 100, thì giảm 3% thành tiền, ngược lại không giảm
  - Tổng tiền phải trả: thành tiền – giảm giá.
8. Viết chương trình hỗ trợ cách giải phương trình bậc 1 ( $ax + b = 0$ )
9. Nhập vào 1 số bất kỳ (0 - 9), cho biết cách đọc số vừa nhập
10. Viết một chương trình dạng menu cho phép người dùng chọn một trong các chức năng bên dưới:
  - In ra màn hình “Xin chào các bạn”
  - Chức năng đổi nhiệt độ C sang độ F
  - Chức năng tính chu vi và diện tích hình tròn

### 2.3. Bài tập về nhà:

1. Giải phương trình bậc 2 có dạng  $ax^2 + bx + c = 0$
2. Nhập vào 3 số nguyên a, b, c. In ra màn hình số nguyên lớn nhất và nhỏ nhất.
3. Nhập vào điểm thi tốt nghiệp THPT (toán, lý, hóa, văn, anh, sử) của một thí sinh. Cho biết điểm trung bình và xếp loại của thí sinh đó, biết rằng xếp loại thi tốt nghiệp được xác định như sau:
  - Điểm trung bình  $\geq 8$  và không có môn nào dưới 6.5: loại giỏi
  - Điểm trung bình  $\geq 7.0$  và không có môn nào dưới 5.0: loại khá
  - Điểm trung bình  $\geq 5.0$  và không có môn nào dưới 3.5: loại TB
  - Còn các trường hợp còn lại xếp loại Yếu.
4. Viết chương trình tính tiền karaoke theo cách sau:

Quy cách tính tiền



5. Viết chương trình tính tiền điện sử dụng trong tháng:
  - Từ 1 – 100KW: 5\$/1KW
  - Từ 101 – 150KW: 7\$/1KW
  - Từ 151 – 200KW: 10\$/1KW
  - Từ 201 – 300KW: 15\$/1KW
  - Từ 300KW trở lên: 20\$/1KW

Ví dụ: Một hộ gia đình có mức tiêu thụ điện là 180KW, tiền điện được tính như sau:

$$5 \cdot 100 + 7 \cdot 50 + 10 \cdot 30 = 1150\$$$

6. Nhập vào 3 số nguyên a, b, c. Kiểm tra xem 3 số đó có là độ dài 3 cạnh một tam giác không? Nếu có thì chúng tạo thành tam giác gì? Gợi ý: 3 số là độ dài 3 cạnh một tam giá nếu như tổng 2 cạnh bất kỳ lớn hơn cạnh còn lại.
7. Viết chương trình tính điểm trung bình cho môn học Thực hành Nhập môn lập trình của sinh viên. Biết rằng môn này có 4 cột điểm, hệ số như nhau. Xuất ra điểm trung bình (hệ 10), đổi điểm sang hệ chữ.
8. Viết chương trình tính điểm trung bình chung trong HK1 của sv ngành CNTT

trường Lý Tự Trọng như sau.

- Nhập vào điểm trung bình tất cả các môn học của HK1, kèm theo số tín chỉ của môn.
- Tính điểm trung bình chung của HK. Xuất ra điểm trung bình chung hệ 10, hệ điểm 4 và điểm chữ.

## CHƯƠNG 3: VÒNG LẶP FOR

**Giới thiệu:** Các nội dung được trình bày về vòng lặp for, vận dụng cú pháp vòng lặp for để giải quyết các bài toán có sự lặp đi lặp lại nhiều lần.

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Nhận biết được công dụng của vòng lặp
- Phát biểu được cú pháp vòng lặp for
- Vận dụng thành thạo vòng lặp for
- Viết chương trình thực thi có sử dụng vòng lặp for
- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

**Nội dung chính:**

### 3.1. Tóm tắt lý thuyết

#### 3.1.1. Khái niệm lặp

- Lặp: là việc thực hiện công việc giống nhau hoặc tương tự nhau nhiều lần
- VD: Cổ động viên Việt Nam hô vang 5 lần “Việt Nam vô địch”

#### 3.1.2. Vòng lặp for

✓ Điều kiện sử dụng:

- Sử dụng khi có công việc lặp xảy ra.
- Biết trước số lần lặp

✓ Cú pháp:

```
for(<biểu thức 1>; <biểu thức 2>; <biểu thức 3>)  
{  
    //Công việc cần lặp  
}
```

VD: Xuất 5 lần câu “Việt Nam vô địch” ra màn hình.

```
for(int i=1; i<=5; i++)  
{  
    printf("Viet Nam vo dich.\n");  
}
```

### 3.2. Bài tập mẫu hướng dẫn thực hành

1. In ra màn hình 10 lần câu “Xin chào các bạn” (mỗi câu nằm trên 1 dòng)

Bước 1: Phân tích bài toán

- Xuất ra 10 lần câu “Xin chào các bạn”. Mỗi câu nằm trên một dòng
- Bài toán có phép lặp với số lần lặp là 10=> sử dụng vòng lặp for
- Công việc cần lặp: In ra màn hình câu nói “Xin chào các bạn”

Bước 2: Cài đặt chương trình

- Khai báo thư viện và viết hàm main

```
#include "stdio.h"
```

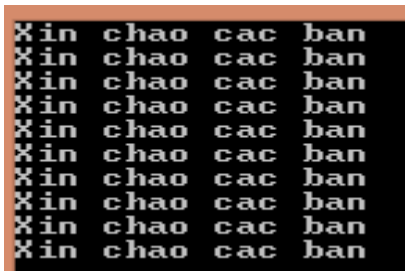
```
#include "conio.h"
```

- Khai báo và nhập dữ liệu: không có

- Cài đặt câu lệnh lặp

```
for(int i=0; i<10; i++)  
{  
printf("\n Xin chhao cac ban");  
}
```

Bước 3: Kiểm tra, sửa lỗi và chạy thử



2. Nhập vào số nguyên dương n. Tính tổng  $S=1^2+2^2+3^2+\dots+n^2$

Bước 1: Phân tích bài toán

- Nhập vào số nguyên dương n
- Tính được tổng bình phương của các số nguyên từ 1 tới n
- 02 biến n, S số nguyên (int)
- Sử dụng vòng lặp for lặp n lần (GV giải thích rõ)
- Câu lệnh lặp:  $S=S+i^2$

Bước 2: Cài đặt chương trình

- Khai báo thư viện và viết hàm main
- Khai báo, nhập dữ liệu

```
int n;  
printf("\n Nhap vao 1 so nguyen duong:"); scanf("%d",  
&n);
```

- Cài đặt câu lệnh lặp

```
int S=0;  
for(int i=1; i<=n; i++)  
{  
  
S = S + i*i;  
}  
printf("\n Tong S=%d",S);
```

Bước 3: Kiểm tra, sửa lỗi và chạy thử

```
nhap vao so nguyen duong n:4
Tong S=30
```

**Lưu ý:** Cách xác định các thành phần của vòng lặp for cho bài toán tính giá trị của một biểu thức:

- Lấy công thức cuối cùng của biểu thức ( $n^2$ ).
- Xác định là biểu thức dùng tính tổng hay tính nhân, thay n bằng i. ta sẽ được công thức lặp ( $i^2 \Rightarrow$  công thức là  $S=S+i^2$ )
- Áp dụng công thức sử dụng i vào số hạng đầu tiên ta sẽ tính được giá trị ban đầu của i

( $i^2=1 \Rightarrow i=1$ )

Ví dụ: Xác định công thức vòng lặp cho biểu thức:  $S=1/1 + 1/2 + 1/3 + \dots + 1/n$ ;

- Công thức cuối:  $1/n$
- Công thức tính tổng:  $S=S+1/i$  (thay n bằng i).
- Giá trị lặp ban đầu của i là  $1(i=1)$  vì  $1/i=1/1$

3. Nhập vào số nguyên dương n. Cho biết số nguyên dương n có bao nhiêu ước là số nguyên dương.

Bước 1: Phân tích bài toán

- Nhập vào số nguyên dương n
- Đếm số ước nguyên dương của n đã nhập
- 02 biến *n*, *dem* là các số nguyên (int)
- Để đếm số ước của n ta cần lần lượt kiểm tra tất cả các số từ 1 tới n xem có phải là ước của n không. Nếu n chia hết cho số nào thì số đó là ước và ta sẽ đếm số đó. Vậy bài toán có công việc lặp với số lần lặp là từ 1 tới n, do đó dùng vòng lặp for
- Câu lệnh lặp: Mỗi lần lặp cần phải kiểm tra xem n có chia hết cho i không. Nếu chia hết thì đếm, câu lệnh lặp có sử dụng lệnh rẽ nhánh if (vì chỉ có một trường hợp rẽ nhánh): nếu  $n\%i==0$  thì  $dem++$

Bước 2: Cài đặt chương trình

- Khai báo thư viện và viết hàm main
- Khai báo, nhập dữ liệu

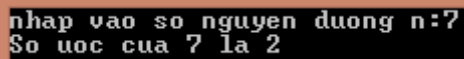
```
int n;
printf("\n Nhập vào 1 số nguyên dương:"); scanf("%d",
&n);
```

- Cài đặt câu lệnh lặp kết hợp rẽ nhánh

```
int dem=0;
for(int i=1; i<=n; i++)
{
    if(n%i==0)
```

```
dem++;  
}  
printf("\n So uoc cua %d la: %d", n, dem);
```

Bước 3: Kiểm tra, sửa lỗi và chạy thử



```
nhap vao so nguyen duong n:?  
So uoc cua 2 la 2
```

### 3.3. Bài tập ở lớp

1. Viết chương trình tính tổng số chẵn nhỏ hơn n, n là số nguyên dương do người dùng nhập từ bàn phím.
  2. Viết chương trình đếm số nguyên dương nhỏ hơn n chia hết cho 3, n là số nguyên dương nhập từ bàn phím.
  3. Viết một chương trình dạng menu cho phép người dùng thực hiện một trong các biểu thức sau:
  4.  $S = 1 * 2 * 3 * 4 * 5 * \dots * n$ .
  5.  $S = 2 + 4 + 6 + 8 + \dots + 20$ .
  6.  $S = 1 * 2 + 2 * 3 + 3 * 4 + 4 * 5 + \dots + n(n+1)$ .
  7.  $S = 1/(1*2*3) + 1/(2*3*4) + 1/(3*4*5) + \dots + 1/(n*(n+1)*(n+2))$
- Xuất tất cả các số nguyên là bội của 7 và có đúng 2 chữ số.
8. Xuất tất cả các số nguyên có đúng 3 chữ số và tổng các chữ số là chẵn.
  9. Bài toán cổ “Trăm trâu trăm cỏ” :

*Trâu đứng ăn năm*

*Trâu nằm ăn ba*

*Lụ khụ trâu già Ba con một bó*

*Trăm con ăn cỏ*

*Trăm bó no nê.*

*Hỏi có bao nhiêu trâu đứng , trâu nằm , trâu già ?*

### 3.4. Bài tập về nhà:

1. Viết chương nhập số nguyên lớn N (khai báo: long N) có k chữ số.
    - Tìm chữ số hàng đầu tiên của N.
    - Tính tổng các chữ số của N.
    - Tìm chữ số lớn nhất trong k chữ số đó.
    - Đếm số chữ số của N.
    - N có phải là số có các chữ số khác nhau đôi một không?
- Gợi ý: Áp dụng các phép toán / và % cho 10 để lấy lần lượt từng chữ số của N theo chiều từ chữ số hàng đơn vị đến các chữ số hàng chục, hàng trăm,...
- Ví dụ:  $N = 347285108$ ,  $x = N \% 10 = 8$ ,  $y = N / 10 = 34728510$ . Lặp lại 2 phép toán chia này ta lần lượt lấy được từng chữ số của N từ hàng đơn vị.
- Khi đó N có:

- Chữ số hàng đầu tiên là 3.
- Tổng các chữ số:  $3+4+7+2+8+5+1+0+8 = 38$

- Chữ số lớn nhất là 8.
- Số chữ số của N: 9
- N không phải số có các chữ số khác nhau đôi một vì có 2 chữ số 8.

2. Nhập vào số nguyên dương h, in ra màn hình các tam giác sau

```

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

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

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

3. Nhập  $n \geq 1$ . Tính  $S(n) = 1 - 2 + 3 - 4 + \dots + (-1)^{n-1}n$ .

4. Nhập  $x, n$ . Tính  $T(x, n) = x^n$

5. Nhập  $x, n$ . Tính  $S(x, n) = x + x^2 + x^3 + \dots + x^n$

6. Nhập  $x, n$ . Tính  $S(x, n) = -x + x^2 - x^3 + \dots + (-1)^n x^n$

7. Viết chương trình tính :

a.  $S = 1 + 3 + 5 + \dots + n$  (n là số dương lẻ)

b.  $S = 2 + 4 + 6 + \dots + 2 \cdot n$

c.  $S = 1/2 + 1/3 + 1/4 + \dots + 1/n$

d.  $S = 1/2 + 3/4 + 5/6 + \dots + n/(n+1)$

e.  $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$

f.  $P = x^n$  (thực hiện bằng 2 cách dùng và không dùng hàm pow())

g.  $P = 1! + 2! + 3! + \dots + n!$

## CHƯƠNG 4: VÒNG LẶP WHILE , DO...WHILE

**Giới thiệu:** Các nội dung được trình bày về vòng lặp while, do...while, vận dụng cú pháp vòng lặp không biết trước số lần lặp để giải quyết các bài toán có sự lặp đi lặp lại nhiều lần.

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Phân biệt được các cấu trúc vòng lặp
- Vận dụng thành thạo vòng lặp while.
- Vận dụng thành thạo vòng lặp do ...while
- Lựa chọn cấu trúc lặp thích hợp cho một số bài toán lập trình.
- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

**Nội dung chính:**

### 4.1. Tóm tắt lý thuyết

#### 4.1.1. Cấu trúc lặp while

–Sử dụng khi bài toán có công việc lặp mà không biết số lần lặp là bao nhiêu. Chỉ xác định được điều kiện dừng lặp của bài toán.

–Ví dụ: Nhập vào dãy số nguyên cho tới khi người dùng nhập số 0 thì dừng. Trong ví dụ, công việc nhập số nguyên lặp đi lặp lại nhiều lần nhưng không biết là lặp bao nhiêu lần. Vòng lặp chỉ dừng khi người dùng nhập số 0.

- Cú pháp:

```
while (<điều kiện>
{
    //công việc lặp
}
```
- Ví dụ:

```
while (x!=0)
{
    printf(" nhap x: ");
    scanf("%d", &x);
}
```

–Trong câu lệnh while, nếu biểu thức điều kiện trả về giá trị đúng thì vòng lặp tiếp tục được thực hiện

–Để tránh vòng lặp vô tận, trong câu lệnh while phải có câu lệnh làm thay đổi giá trị của biểu thức điều kiện.

#### 4.1.2. Cấu trúc lặp do...while

- Tương tự như cấu trúc lặp while
- Thực hiện ít nhất một lần (while có thể không thực hiện lần nào nếu ngay mới lần đầu tiên biểu thức điều kiện trong while trả về giá trị sai)

- Cú pháp:

do

{

    //công việc cần lặp

} while (<điều kiện>);

- Ta có thể dùng từ khóa break để thoát ra khỏi vòng lặp

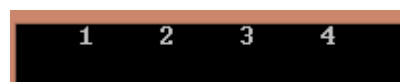
int i=1;

do

{

    if(i==5) break; printf("%d", i); i++;

}while(i<10);



1 2 3 4

- Dùng từ khóa continue để chuyển sang bước lặp tiếp theo.

int i=1;

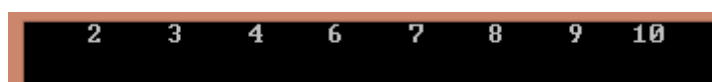
do

{

    if(i==5) continue;

    printf("%d", i); i++;

}while(i<10);



2 3 4 6 7 8 9 10

## 4.2. Bài tập mẫu hướng dẫn thực hành

1. Nhập vào 2 số nguyên dương a và b. Tìm ước chung lớn nhất của 2 số nguyên dương đã nhập.

Bước 1: Phân tích bài toán

- Nhập 2 số nguyên bất kỳ
- Xuất ra ước chung lớn nhất của 2 số vừa nhập
- 02 biến số nguyên (int), 01 biến ucln (int)
- Sử dụng thuật toán Euclid => số lần lặp của bài toán không xác định  
=> sử dụng vòng lặp while
- Thuật toán Euclid tìm UCLN của a và b  
Nếu  $a=b$  thì  $UCLN = a$ ;  
Nếu  $a > b$  thì  $UCLN(a,b) = UCLN(a-b,b)$   
Nếu  $a < b$  thì  $UCLN(a,b) = UCLN(a,b-a)$

Bước 2: Cài đặt chương trình

- Khai báo thư viện và viết hàm main
- Khai báo và nhập dữ liệu

```
int a,b;

printf("\n nhap vao 2 so a va b:");
scanf("%d%d",&a,&b);
```
- Cài đặt câu lệnh lặp

```
while (a!=b)
{
    if(a>b)
        a = a- b;
    else
        b = b - a;
}

printf("UCLN la: %d", a);
```

Bước 3: Kiểm tra, sửa lỗi và chạy thử

2. Xây dựng một menu chương trình cho phép người dùng lựa chọn gồm các yêu cầu sau:

1. In ra màn hình từ hello
2. In ra màn hình câu Vòng lặp for
3. In ra màn hình câu Vòng lặp while
4. Thoát chương trình.

Sau khi lựa chọn chức năng, chương trình sẽ thực hiện theo đúng chức năng đã

chọn rồi lại tiếp tục hiển thị menu. Chương trình chỉ dừng lại (kết thúc khi người dùng bấm phím 4)

Bước 1: Phân tích bài toán

- Xuất ra màn hình menu theo yêu cầu
- Sử dụng 01 biến luachon là số nguyên (int)
- Menu này xuất hiện ít nhất 1 lần (cho lần chọn đầu tiên) và lặp đi lặp

lại nhiều lần => dùng cấu trúc lặp do...while

Bước 2: Cài đặt chương trình

- Cài đặt chương trình

```
int luachon;

do{

    printf("1. In ra màn hình tu Hello. \n");
    printf("2. In ra màn hình Vòng lặp for \n");
    printf("3. In ra màn hình Vòng lặp while \n");
    printf("4. Thoát chương trình \n");
    printf("Bạn chọn chức năng nào: \n");
    scanf("%d", &luachon);
    switch(luachon)

    {

        case 1: printf("HELLO \n"); break;
        case 2: printf("Vòng lặp FOR \n"); break;
        case 3: printf("Vòng lặp WHILE \n"); break;
        case 4: printf("Thoát chương trình\n"); break;

    }

} while (luachon!=4);
```

Bước 3: Kiểm tra, sửa lỗi và chạy thử

3. Viết chương trình nhập vào một số nguyên dương n. Tính tổng các chữ số có trong số nguyên dương n

Bước 1: Phân tích bài toán

- Nhập vào số nguyên dương n
- Tính tổng các chữ số có trong số nguyên dương n
- Cách thực hiện, ví dụ với n=629
  - tong=0; n=629>0:
  - ✓ tong=tong + hàng đơn vị của n (số 9: n%10)
  - ✓ Gán lại giá trị
  - cho n: n=n/10
  - n=62>0:

✓ tong=tong + hàng đơn vị của n (số 2:  $n\%10$ )

✓ Gán lại giá trị cho n:  $n=n/10$

$n=6>0$ :

✓ tong=tong + hàng đơn vị của n (số 6:  $n\%10$ )

✓ Gán lại giá trị

cho n:  $n=n/10$

$n=0$ :dừng

- Ta thấy việc tong=tong+hàng đơn vị của n và  $n=n/10$  được lặp lại nhiều lần nhưng không biết cụ thể bao nhiêu lần. Do đó ta phải dùng vòng lặp while.

#### Bước 2: Cài đặt chương trình

- Khai báo và nhập dữ liệu:

```
int n;
```

```
printf("\n Nhập vào 1 số nguyên dương:");
```

```
scanf("%d", &n);
```

- Cài đặt câu lệnh `int tong=0; while(n>0)`

```
{
```

```
    tong = tong
```

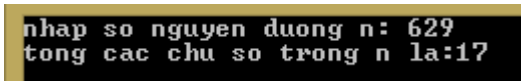
```
    + n%10;
```

```
    n=n/10;
```

```
}
```

```
printf("tong các chữ số trong n là:%d", tong);
```

#### Bước 3: Kiểm tra, sửa lỗi và chạy thử



```
nhap so nguyen duong n: 629
tong cac chu so trong n la:17
```

### 4.3. Bài tập ở lớp

1. Viết chương trình kiểm tra xem n có phải là số nguyên tố hay không. Biết rằng số nguyên tố là số nguyên lớn hơn 1 và chỉ có 2 ước là 1 và chính nó.
2. Viết chương trình cho biết trong số nguyên n có bao nhiêu chữ số chẵn.
3. Viết chương trình tìm bội chung nhỏ nhất (BCNN) của 2 số nguyên dương a và b nhập vào từ bàn phím. Biết rằng  $BCNN(a,b)=a*b/UCLN(a,b)$ .
4. Viết chương trình thực hiện việc lặp lại thao tác nhập và xuất một số nguyên bất kỳ từ bàn phím. Việc nhập kết thúc khi người dùng nhập vào giá trị 0. Hiển thị tổng các số vừa nhập.
5. Viết chương nhập một số nguyên không âm và ra số đảo ngược. Ví dụ, nếu nhập vào 483 thì in ra 384.

6. Viết chương trình tính tổng và tích của dãy số gồm  $n$  phần tử. Trong đó số  $n$  và các phần tử được nhập vào từ bàn phím khi thực hiện chương trình.

7. Viết chương trình kiểm tra và in ra các số nguyên tố nhỏ hơn 100

8. Viết chương trình cho phép người nhập vào  $n$  số thực bất kỳ với  $n$  là giá trị nhập. Sau đó thực hiện các công việc sau:

- Đếm xem có bao nhiêu số thực dương
- Tính tổng các số thực dương
- Tính trung bình cộng của các số thực dương
- Tìm số lớn nhất và số nhỏ nhất trong tất cả các số được nhập
- Tìm số lớn nhất và nhỏ nhất trong các số thực dương

9. Viết chương trình cho phép người nhập vào một dãy các số nguyên dương, kết thúc nhập khi người dùng nhập vào số 0. Sau đó thực hiện các công việc sau:

- Đếm xem có bao nhiêu số chẵn
- Tính tổng các số chẵn
- Tính trung bình cộng của các số chẵn
- Tìm số lớn nhất và số nhỏ nhất trong tất cả các số chẵn
- Đếm xem có bao nhiêu số nguyên tố

10. Viết chương trình thực hiện việc lặp lại thao tác nhập và xuất một số nguyên có đúng 3 chữ số và các chữ số phải khác nhau từ bàn phím. Việc nhập kết thúc khi người dùng nhập vào giá trị vi phạm điều kiện trên. Hiển thị số lượng các số vừa nhập.

#### 4.4. Bài tập về nhà:

1. Kiểm tra số nguyên dương  $n$  có các chữ số toàn là chữ số chẵn hay không?
2. Kiểm tra số nguyên dương  $n$  có các chữ số toàn là chữ số lẻ hay không?
3. Kiểm tra số nguyên dương  $n$  có các chữ số tăng dần từ trái qua phải không?
4. Kiểm tra số nguyên dương  $n$  có các chữ số giảm dần từ trái qua phải không?
5. Viết chương trình nhập một số nguyên dương  $n$ . Tìm số nguyên dương  $m$  lớn nhất sao cho  $1+2+3+\dots+m < n$ .
6. Viết chương trình nhập một số nguyên dương  $n$ . Tìm số nguyên dương  $m$  nhỏ nhất sao cho  $1+2+3+\dots+m > n$ .
7. Viết chương trình xuất số đảo của số nguyên dương  $n$ ,  $n$  nhập từ bàn phím.
8. An gửi số tiền là  $n$  USD vào ngân hàng. An muốn số tiền của mình sẽ bằng hoặc lớn hơn 1000 USD. Giúp An viết chương trình tính số tháng cần gửi

vào ngân hàng để tổng tiền vốn lẫn tiền lãi sẽ lớn hơn 1000USD. Biết lãi suất 0.7%/ tháng và số tiền vốn n được nhập từ bàn phím.

## CHƯƠNG 5: HÀM TRONG C/C++

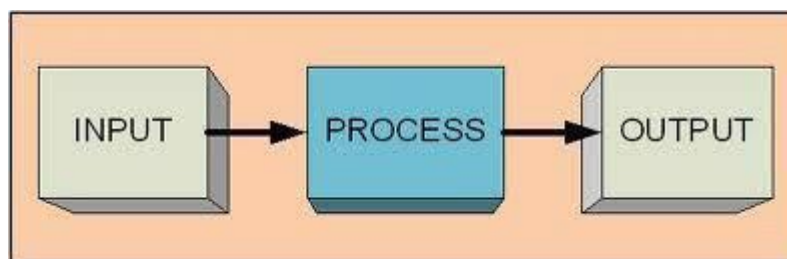
**Giới thiệu:** Giới thiệu về cách viết hàm, phân rã chương trình thành những hàm con, sử dụng hàm đúng cú pháp. Truyền tham số và khai báo tham số đúng mục đích gồm tham biến và tham trị. Cài đặt hàm đệ quy

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Xác định được cấu trúc và các thành phần của hàm
- Áp dụng được cấu trúc lặp while, do...while để xây dựng các hàm.
- Phân tích được yêu cầu bài toán và phân rã thành hàm con
- Cài đặt được hàm bằng ngôn ngữ C/C++.
- Truyền tham số và sử dụng hàm một cách chính xác
- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

**Nội dung chính:**

### 5.1. Tóm tắt lý thuyết



#### 5.1.1. Các thành phần của hàm:

Hàm là một đoạn chương trình dùng để giải quyết một vấn đề cụ thể nào đó. Một hàm gồm các thành phần sau:

- Tên hàm: do người lập trình tự đặt theo qui tắc đặt tên của C++
- Input: đầu vào của hàm
- Output: đầu ra của hàm
- Processing: quá trình xử lý của hàm (thuật toán)

Ví dụ: Viết hàm tính diện tích hình tròn khi biết bán kính

- Tên hàm: `TinhDienTichHinhTron`
- Input: bán kính
- Output: diện tích hình tròn
- Processing:  $S = \text{bán kính} * \text{bán kính} * \text{PI}$   
(Trong đó, PI là hằng số có giá trị là 3.1416)

#### 5.1.2. Cú pháp khai báo hàm

**kiểu trả về** <Tên hàm> (<danh sách tham số>)

```
{
    //Câu lệnh thân hàm
```

Trong đó

- <kiểu trả về>: kiểu dữ liệu của output, nếu hàm không có output thì kiểu trả về là void
- <danh sách tham số>: là tất cả các input của hàm (bao gồm cả kiểu dữ liệu của input và tên của input, các input cách nhau bởi dấu “,”)
- Thân hàm: thuật toán (các câu lệnh bắt đầu từ input tìm ra output)

Ví dụ: Viết hàm tính diện tích hình tròn khi biết bán kính

```
float DTHinhTron(float bankinh)
{
    float S;
    S =
    bankinh*bankinh*3.
    14; return S;
}
```

Ví dụ: In ra màn hình n lần từ Hello, mỗi từ trên 1 dòng

```
void InHello (int n)
{
    for(int i=0; i<=n; i++)
        printf("Hello\n");
}
```

### 5.1.3. Cách gọi hàm

- Ta có thể gọi hàm ở bất kỳ vị trí nào, trong chương trình chính hoặc trong thân của một hàm khác
- Hàm phải được khai báo ở vị trí sau phần khai báo thư viện (prototype)
- Cách gọi hàm đối với hàm có kiểu trả về void

**<Tên hàm> (<ds tham số>);**

**Ví dụ:** Muốn in ra màn hình 10 lần từ Hello ta gọi hàm như sau:

```
InHello (10);
```

Muốn in ra màn hình 25 lần từ Hello ta gọi hàm như sau:

```
InHello (25);
```

- Cách gọi hàm đối với hàm có kiểu trả về **khác void**

Khai báo biến có cùng kiểu trả về với hàm Gán giá trị của hàm vào cho biến vừa khai báo

Xử lý biến theo ý muốn

<tên biến> =<Tên hàm>(ds tham số);

Ví dụ: ta có bán kính hình tròn  $r=6$ . Cần tính diện tích hình tròn rồi in diện tích đó ra màn hình ta có thể gọi như sau:

```
int r=6;
float dt = TinhDienTichHinhTron(r);
printf("Diện tích hình tròn là: %.2f", dt);
```

## 5.2. Bài tập mẫu hướng dẫn thực hành

1. Viết hàm tìm ước chung lớn nhất của 2 số nguyên dương a và b. Viết hàm main tính ước chung lớn nhất cho 2 số nguyên nhập vào từ bàn phím, sử dụng hàm đã viết ở trên

Bước 1: Xác định tên hàm, input, output, thuật toán

- Tên hàm: UCLN (ước chung lớn nhất)
- Input: 2 số nguyên a, b (kiểu int)
- Output: ước chung lớn nhất của a,b là 1 số nguyên
- Sử dụng thuật toán euclid => số lần lặp của bài toán không xác định  
=> sử dụng vòng lặp while
- Thuật toán euclid tìm
  - UCLN của a và b
  - Nếu  $a=b$  thì UCLN
  - $=a$ ;
  - Nếu  $a>b$  thì
  - $UCLN(a,b)=UCLN(a-b,b)$
  - Nếu  $a<b$  thì
  - $UCLN(a,b)=UCLN(a,b-a)$

Bước 2: Cài đặt chương trình

- Cài đặt hàm

```
int UCLN(int a, int b)
{
    while (a!=b)
        if(a>b)
            a = a - b;
        else
```

```

        b = b - a;

    return a;

}

- Gọi hàm
  int x,y;
  printf("\n nhap vao 2 so nguyen duong:");
  scanf("%d%d",&x,&y);
  int uc = UCLN(x,y);
  printf("Uoc chung lon nhat can tim la:%d", uc);

```

2. Viết hàm kiểm tra số nguyên dương n có phải là số nguyên tố hay không? Viết hàm main minh họa kiểm tra số nguyên nhập vào từ bàn phím có phải là số nguyên tố hay không?

Bước 1: Xác định tên hàm, input, output, thuật toán

- Tên hàm: ktraSNT (kiểm tra số nguyên tố)
- Input: 01 số nguyên dương n (kiểu int)
- Output: có hoặc không, đúng hoặc sai, true hoặc false => kiểu trả về là int (1: đúng, 0: sai)
- Thuật toán: i chạy từ 2 tới sqrt(n)  
 Nếu n chia hết cho i nào đó thì đó không là số nguyên tố  
 Nếu không có giá trị i nào để n chia hết cho i thì n là số nguyên tố

Bước 2: Cài đặt chương trình

```

- Cài đặt hàm
  int KTSNT(int n)
  {
      if(n<2)
          return 0; //khong la so nguyen to

      for(int i=2; i<=sqrt(float)n; i++)
          if(n%i==0)
              return 0;

      return 1; //la so nguyen to
  }

- Gọi hàm
  int so,kq;
  printf("\n nhap vao so nguyen muon kiem tra:");
  scanf("%d",&so);

```

```
kq = KTSNT(so); if(kq==1)
    printf("\n Day la so nguyen to");
else
    printf("\n Day khong phai la so nguyen to");
```

3. Viết hàm xuất ra các số lẻ từ n về 1

Bước 1: Xác định tên hàm, input, output, thuật toán

- Tên hàm: XuatLeTuNVe1
- Input: 01 số nguyên dương n (kiểu int)
- Output: xuất ra màn hình => kiểu trả về là void
- Thuật toán: i chạy từ n về 1  
Nếu i lẻ thì xuất i ra màn hình

Bước 2: Cài đặt chương trình

```
void XuatLeTuNVe1(int n)
{
    for(i=?;i ?;i--)
        if(.....)
            printf("%d\t",i);
}
```

Viết hàm main để thực thi

### 5.3. Bài tập ở lớp

1. Viết chương trình tính diện tích hình tam giác khi biết ba cạnh của tam giác (yêu cầu viết hàm tính diện tích tam giác)
2. Viết chương trình xuất ra màn hình một thực đơn cho phép chọn 1 trong 2 tác vụ:

Nhập tổng số thời gian tính bằng giây, đổi sang dạng: hour:minute:second

Nhập vào thời gian dạng hour:minute:second, đổi sang thời gian bằng giây

Trong chương trình, cần thiết kế các hàm sau:

```
int Menu(); //Xuất thực đơn và chọn mục trong thực đơn
```

```
void Sec2Time(long sec); //Chuyển số giây sang dạng thời gian
```

```
long Time2Sec(int hour, int minute, int second); //Chuyển thời gian sang số giây
```

3. Hãy viết một chương trình có gọi sử dụng các hàm sau:

- a. Hàm tính n!

```
long Giaithua(int n);
```

- b. Hàm hoán vị 2 số

**void Swap(float &a, float &b);**

- c. Hàm in n dấu \*

**void InSao(int n);**

- d. Hàm kiểm tra số nguyên tố.

**int KTNTTo(int n); // trả về 1 nếu n nguyên tố, ngược lại trả về 0**

- e. Hàm đếm số chữ số của số n.

**int SoChuSo(int n); // ví dụ số 231 có 3 chữ số**

- f. Hàm tìm ước số chung lớn nhất của 2 số.

**int UCLN(int a, int b);**

4. Viết chương trình cho phép người dùng thực hiện một trong các yêu cầu sau:

1. Tính tổng các ước của số nguyên n.
2. Đổi chỗ 2 số nguyên x, y.
3. Tìm BCNN của hai số nguyên dương a và b
4. Kiểm tra số nguyên dương n có các chữ số toàn là *chữ số lẻ* hay không?

(Yêu cầu: Mỗi chức năng viết hàm thực hiện, hàm main gọi hàm vừa viết thực thi)

5. Viết hàm tính tổng các số nguyên tố nhỏ hơn n, trong đó n là số nguyên dương nhập từ bàn phím (yêu cầu viết hàm kiểm tra số nguyên tố, hàm tính tổng các số nguyên tố nhỏ hơn n).
6. Viết hàm tính đếm các số chính phương nhỏ hơn n, trong đó n là số nguyên dương nhập từ bàn phím (yêu cầu viết hàm kiểm tra số chính phương, hàm tính đếm các số chính phương nhỏ hơn n).

#### 5.4. Bài tập về nhà:

1. Viết hàm xuất số đảo của số nguyên dương n, n nhập từ phím. Viết hàm chương trình thực thi
2. Nhập vào số nguyên dương h, viết chương trình in ra màn hình một trong các tam giác sau (người dùng tùy chọn)



3. Viết chương trình tính tổng  $S(n)=1-2+3-4+ \dots +(-1)^{n-1}n$ , trong đó  $n$  là số nguyên dương,  $n$  nhập từ bàn phím.

4. Viết chương trình tính tổng sau.

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

5. Viết chương trình tính tổng  $S(n) = 1/3 + 1/5 + \dots + 1/(2*n+1)$

6. Viết hàm kiểm tra một số nguyên  $x$  có là số hoàn thiện (còn gọi là số hoàn chỉnh, số hoàn hảo) không? Biết rằng số hoàn thiện là số có tổng các ước của nó (trừ chính nó) bằng chính nó.

Ví dụ: 6 có các ước 1, 2, 3 và  $1+2+3 = 6$  nên 6 là số hoàn thiện.

7. Xuất  $n$  số nguyên tố đầu tiên với  $n>0$  nhập từ bàn phím.

Ví dụ:  $n=5$ . 5 số nguyên tố đầu tiên là: 2 3 5 7 11

8. Xuất  $n$  số chính phương đầu tiên với  $n>0$  nhập từ bàn phím. Ví dụ:  $n = 3$ .

3 số chính phương đầu tiên là: 1 4, 9

9. Nhập vào 3 phân số dạng  $a/b$ ,  $c/d$ ,  $e/f$  (với  $a,b,c,d,e,f$  là các số nguyên).

Kiểm tra các mẫu số phải khác 0, nếu vi phạm yêu cầu nhập lại. Sau đó tính tổng, tích của 3 phân số với kết quả tối giản.

10. Tìm  $m$  số nguyên tố cùng nhau với số  $n$ , với  $n, m>0$  được nhập từ bàn phím. Biết rằng 2 số nguyên tố cùng nhau khi ước chung lớn nhất của chúng là 1.

## CHƯƠNG 6: TRUYỀN THAM SỐ - HÀM ĐỆ QUY

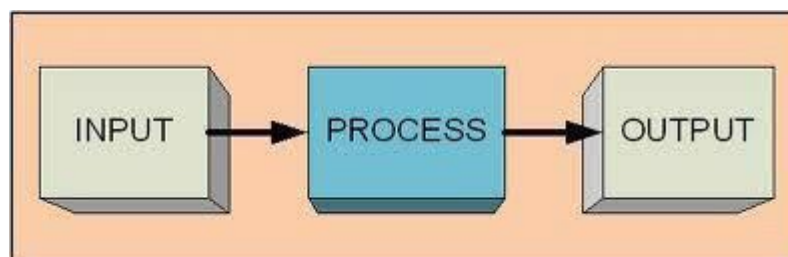
**Giới thiệu:** Phân biệt tham số hình thức, tham số thực, truyền tham số tham trị và truyền tham số tham biến, Khi nào cần truyền tham biến hay tham trị. Truyền tham số và khai báo tham số đúng mục đích gồm tham biến và tham trị. Cài đặt hàm đệ quy

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Xác định được tham số thực, tham số hình thức, tham biến, tham trị
- Vận dụng được vào bài toán và phân biệt khi nào truyền tham số dùng tham biến, khi nào dùng tham trị
- Phân tích được yêu cầu bài toán
- Xác định được cấu trúc hàm đệ quy, các thành phần liên quan
- Cài đặt được hàm bằng ngôn ngữ C/C++.
- Truyền tham số và sử dụng hàm một cách chính xác
- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

**Nội dung chính:**

### 6.1. Tóm tắt lý thuyết



#### 6.1.1. Tham số thực (real parameters)

Là tham số được truyền thực sự cho hàm thông qua lời gọi hàm. Với tham số thực ta không chỉ quan tâm đến kiểu dữ liệu của tham số mà còn quan tâm đến tên của tham số tương ứng cần truyền. Tên của tham số thực do vậy có thể khác với tên của tham số hình thức, và trong nhiều trường hợp tham số thực không phải là tên mà là một hằng cụ thể.

#### 6.1.2. Tham số hình thức (formal parameters)

Là tham số được khai báo trong prototype mô tả hàm. Đối với tham số hình thức điều ta cần quan tâm không phải là tên của tham số mà là kiểu của tham

số cần truyền cho hàm và thứ tự các tham số ghi trong mô tả hàm. Như vậy tên của các tham số trong trường hợp này chỉ mang tính hình thức và chúng có thể khác tên với các tham số sẽ được truyền thực sự cho hàm thông qua lời gọi hàm.

#### 6.1.3. Tham số hình thức trị hay tham trị (value parameter)

Là tham số được truyền cho hàm bằng giá trị của tham số thực. Khi đó sự thay đổi giá trị của tham số trong hàm nếu có sẽ không làm thay đổi giá trị của tham số thực đó trong hàm gọi.

ví dụ với hàm : **long sqr(int x);**

thì **x** là tham trị, vì trong hàm **main()** giá trị của biến **num** là tham số thực truyền cho hàm **sqr()** qua phép gọi hàm **square=sqr (num)** không bị thay đổi sau khi kết thúc việc thực hiện hàm này.

#### 6.1.4. Tham số hình thức biến hay tham biến (variable parameter):

Là tham số được truyền cho hàm bằng địa chỉ của tham số thực. Khi đó sự thay đổi giá trị tương ứng của tham số trong hàm (nếu có) sẽ làm thay đổi giá trị của tham số thực đó ở ngoài hàm.

Trong ví dụ sau với hàm : **void Swap(int &a, int &b);**

Thì **a** và **b** là các tham biến vì tham số được truyền cho hàm là **&a** (địa chỉ của **a**), **&b** (địa chỉ của **b**) nên sự thay đổi giá trị của **a**, **b** bên trong hàm **Swap()** sẽ được cập nhật ngay lên địa chỉ của **a**, **b** cũng chính là địa chỉ của tham số thực của **m** và **n** của hàm **main()** nên giá trị của **m**, **n** sẽ bị thay đổi theo.

#### 6.1.5. Cấu trúc của hàm đệ quy

Hàm đệ quy phải bao gồm 2 thành phần: thành phần dừng và thành phần đệ quy.

Thành phần dừng xác định điểm dừng của sự đệ quy, hay nói cách khác, thành phần dừng là điều kiện để kết thúc giải thuật và trong đó không bao gồm lời gọi hàm tới chính nó.

Thành phần đệ quy là thành phần mà trong đó một hàm sẽ được gọi bởi chính nó để có thể tìm ra giá trị cho các bước đệ quy sau.

### 6.2. Bài tập mẫu hướng dẫn thực hành

1. Viết hàm nhập giá trị nguyên n

- Cài đặt hàm: cách 1

```
int NhapSoNguyen()  
{  
    int a;  
    printf("Nhap a:");  
    scanf("%d",&a);  
    return a;  
}
```

Gọi hàm: num = NhapSoNguyen();

- Cách 2: dùng tham biến

```
void NhapSoNguyen(int &a)  
{  
    printf("Nhap a:");  
    scanf("%d",&a);  
    return a;  
}
```

Gọi hàm: NhapSoNguyen(num);

2. Viết hàm hoán vị hai số nguyên a,b

Bước 1: Xác định tên hàm, input, output, thuật toán

- Tên hàm: Swap
- Input: 2 số nguyên n (kiểu int)
- Output: giá trị hai số nguyên được hoán đổi cho nhau
- Thuật toán:  $t = a; a = b; b = t$

(Chú ý: a, b trong hàm Swap được khai báo là tham biến, tại sao?)

Bước 2: Cài đặt chương trình

- Cài đặt hàm

```
void Swap(int &a, int &b)  
{  
    int t = a;  
    a = b;  
    b = t;  
}
```

```
}
```

- Gọi hàm

```
int a,b;
```

```
printf("\n nhập vào số nguyên a:");
```

```
scanf("%d",&a);
```

```
printf("\n nhập vào số nguyên b:");
```

```
scanf("%d",&b);
```

```
Swap(a,b);
```

```
printf("\n Sau khi hoán vị a = %d, b = %d",a,b);
```

### 3. Viết hàm đệ quy tính $S(n) = 1.2 + 2.3 + 3.4 + \dots + n*(n+1)$

Bước 1: Xác định tên hàm, input, output, thuật toán

- Tên hàm: S
- Input: 01 số nguyên n (kiểu int)
- Output: Giá trị tổng => kiểu trả về là int
- Thuật toán:
  - Thành phần dừng:  $S(1) = 2$ ;
  - Thành phần đệ quy:  $S(n) = S(n-1) + n*(n-1)$

Bước 2: Cài đặt chương trình

```
int S(int n)
```

```
{  
    if(n==1)  
        return ?;  
    else  
        return ?;  
}
```

Viết hàm main để thực thi

## 6.3. Bài tập ở lớp

### 1. Viết hàm tính số Fibonacci thứ n

$F(1)=1; F(2)=1; F(n)= F(n-1)+F(n-2);$

### 2. Viết các hàm sau dùng giải thuật đệ quy :

- Tính  $S_n = n + S_{n-1}$
- Hàm in n dấu \*
- Hàm đếm số chữ số của số n.
- Hàm tìm ước số chung lớn nhất của 2 số.

### 3. Viết các hàm sau bằng 2 cách đệ quy và không đệ quy :

- a. Hàm in ra màn hình các giá trị giảm dần từ n đến 1.
- b. Hàm tính tổ hợp chập k của n phần tử  

$$C(n,k) = C(n-1,k-1) + C(n-1,k);$$
 Với  $C(n,k)=1$  khi  $n=k$  hay  $k=0$ ;  
 Hoặc  $C(n,k) = n!/(k!*(n-k)!)$
4. Viết các hàm sau bằng cách đệ quy :
  - a. Đổi một số thập phân sang số nhị phân
  - b. Đổi một số thập phân sang số bát phân

#### 6.4. Bài tập về nhà:

1. Giả sử lãi suất tiết kiệm kỳ hạn 1 tháng là 5 % tháng (không phân biệt số ngày của tháng, không cho phép rút tiền trước hạn). Sau mỗi tháng nếu người gửi không rút lãi thì lãi sẽ được nhập vốn để tính cho tháng sau. Hãy viết chương trình cho phép người dùng nhập vào số tiền gửi, số tháng gửi và in ra số tiền nhận được gồm cả vốn và lãi. Yêu cầu viết hàm tính bằng 2 cách đệ quy và không đệ quy.

Ví dụ:

Số tiền gửi (đơn vị tính 1.000 VND) = 100000

Thời gian gửi: 3 tháng

Số tiền nhận được (đơn vị tính 1.000 VND): 101255.2

2. Viết hàm tính  $x^n$  bằng hàm sử dụng đệ quy.
3. Viết hàm tính tổng chữ số của số nguyên dương n bằng đệ quy
4. Viết hàm xuất các chữ số chẵn có trong n bằng đệ quy
5. Viết hàm đệ quy và không đệ quy tính số sau  

$$K(n) = 2*K(n-1) + 3*K(n-2)+1, K(1) = 1, K(2) = 3;$$

## CHƯƠNG 7: MẢNG MỘT CHIỀU

**Giới thiệu:** Cung cấp kiểu dữ liệu mảng 1 chiều cho người học. Cách khai báo và sử dụng mảng 1 chiều. Duyệt mảng một chiều và các thao tác trên mảng 1 chiều

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Khai báo được cấu trúc dữ liệu mảng 1 chiều.
- Phân tích được tình huống sử dụng mảng 1 chiều để lưu trữ dữ liệu.
- Cài đặt được các hàm nhập, xuất, xử lý dữ liệu trong các thao tác tính tổng, đếm

giá trị phần tử.

- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

### Nội dung chính:

## 7.1. Tóm tắt lý thuyết

### 7.1.1. Định nghĩa và khai báo mảng 1 chiều:

- Khái niệm:

Mảng một chiều là một nhóm các phần tử có cùng kiểu dữ liệu và cùng một tên, các phần tử trong mảng được lưu liên tiếp với nhau trong bộ nhớ máy tính.

- Khai báo:

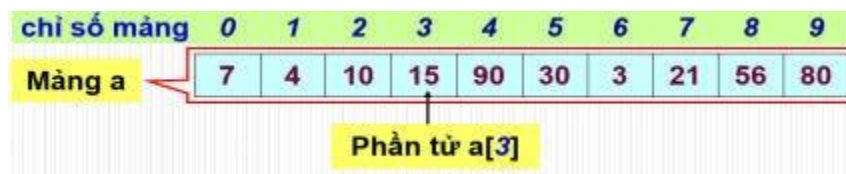
<Kiểu dữ liệu> <Tên mảng>[<Số phần tử tối đa của mảng>];

**Lưu ý:** Chỉ số mảng luôn luôn bắt đầu từ 0.

### **Ví dụ:**

- `int a[100];` //Mảng có tên a chứa tối đa 100 số nguyên
- `#define maxN 30`  
`float b[maxN];` // Mảng có tên b chứa tối đa 30 số thực
- `const int maxC = 50`  
`char c[maxC];` Mảng có tên c chứa tối đa 50 kí tự

**Ví dụ:** Cho mảng 1 chiều tên a chứa 10 số nguyên như sau:



Trong đó:

- i là chỉ số của mảng ( $i \in [0, 9]$ ).
- $a[i]$  là giá trị của phần tử thứ i+1 (chỉ số i)

### 7.1.2. Tham số mảng của một hàm

```

<Kiểu dữ liệu> <Tên hàm> (<kiểu dữ liệu mảng> <Tên mảng>[], int <ten so phan tu>)
{
    //Khởi tạo các biến
    //Duyệt mảng để tính toán
    //Trả về giá trị
}

```

### 7.1.3. Những thao tác trên mảng:

Xử lý mảng là việc xử lý nhóm các phần tử trong mảng lần lượt từng phần tử một.

- Duyệt theo chiều xuôi

Duyệt mảng không có điều kiện:

```

for(int i=0; i<n; i++)
{
    //xử lý a[i];
}

```

Ví dụ:

```

for(int i=0; i<5; i++)
    printf("%5d",
        a[i]);

```

Duyệt mảng có điều kiện

```

for(int i=0; i<n; i++)
{
    if(<Điều kiện>)
        //xử lý a[i];
}

```

Ví dụ: `int S=0;`

```

for(int i=0; i<5; i++)
    if(a[i]%2==0)
        S = S + a[i];

printf("\n Tổng các số chẵn trong mảng là: %d", S);

```

- Duyệt theo chiều ngược

Duyệt mảng không có điều kiện:

```

for(int i=n-1; i>=0; i--)

```

```
{
    //Xử lý a[i]
}
```

Duyệt mảng có điều kiện:

```
for(int i=n-1;i>=0;i--)
{
    if(<điều kiện>)
        //Xử lý a[i]
}
```

## 7.2. Bài tập mẫu hướng dẫn thực hành.

1. Viết chương trình nhập các giá trị cho mảng 1 chiều chứa các số nguyên, sau đó xuất mảng vừa nhập (Lưu ý: Viết chương trình dạng hàm).

Yêu cầu:

- Viết 2 hàm Nhập và Xuất mảng 1 chiều chứa các số nguyên.
- Gọi thực hiện 2 hàm này trong hàm main. Hướng dẫn:

- Khai báo thêm các thư viện cơ bản cho chương trình.

```
#include <conio.h>
```

```
#include <stdio.h>
```

- Khai báo hằng số cho chương trình.

```
#define MAXSIZE 1000
```

- Viết các định nghĩa hàm nhập mảng, xuất mảng của chương trình như sau:

```
void NhapMang1C_SoNguyen(int a[], int &n)
{
    do
    {
        printf("Cho biet so phan tu cua mang: ");
        scanf("%d", &n);
    }while(n<=0);
    for(int i=0; i<n; i++)
    {
        printf("Gia tri phan tu a[%d]=",
            i); scanf("%d", &a[i]);
    }
}
```

```

    }
}

//=====

void XuatMang1C_SoNguyen(int a[], int n)
{
    for(int i=0; i<n; i++)
        printf("%4d", a[i]);
}

```

- Viết hàm main để thực thi chương trình.

```

void main()
{
    int a[MAXSIZE];
    int n; NhapMang1C_SoNguyen(a, n);
    printf("\nNoi dung cua mang la: ");
    XuatMang1C_SoNguyen(a, n);
    getch();
}

```

2. Bổ sung vào bài 1 các thao tác sau (Lưu ý: Viết chương trình dạng hàm):

- **Xuất các giá trị chẵn của mảng ra màn hình.**
- **Tính tổng các giá trị mảng.**
- **Tính tổng chẵn trong mảng**

Hướng dẫn:

- Bổ sung các định nghĩa hàm cần thiết cho chương trình như sau:

```

//=====
====

void Xuat_SoChan(int a[], int n)
{
    // SV tự làm
}

int TongMang (int a[], int n)
{
    // SV tự làm }

```

```
int TongChan (int a[], int n)
{
    // SV tự làm
}
```

- Bổ sung vào hàm main để thực thi chương trình.

```
void main()
{
    int A[MAXSIZE], N;
    //*****
    TaoMang1C_SoNguyen(A, N);
    printf("\nNoi dung cua mang vua tao la:");
    XuatMang1C_SoNguyen(A,N);
    //*****
    Xuat_SoChan(A, N);
    int t = TongMang(A,N);
    printf("\Tong cua mang
    la:%d", t); int tc =
    TongChan(A,N);
    printf("\Tong cua cac gia tri chan trong mang
    la:%d", tc);
}
```

### 7.3. Bài tập ở lớp

Cho một mảng 1 chiều chứa các số nguyên. Hãy viết một chương trình cho phép người dùng chọn một trong các yêu cầu sau:

1. Viết chương trình thực hiện các chức năng sau (mỗi chức năng viết một hàm riêng):

- a. Nhập mảng các số thực, kết thúc nhập khi người dùng nhập 9999.
- b. Xuất mảng các số thực.
- c. Tính trung bình cộng các số thực có trong mảng.
- d. In ra 2 dòng, một dòng gồm các số thực dương, một dòng gồm các số thực âm.
- e. Đếm số lần xuất hiện của số x.  
Ví dụ với dãy số: 1.0, -3.5, 6.7, 1.0, 8.7, 12.32, 0.58, 6.7, 12.5, -1.0, 6.7, 4.15

thì số lần xuất hiện của x=6.7 là 3.

- f. Tìm giá trị min của mảng
- g. Xóa tất cả các phần tử có giá trị bằng x trong mảng.
- h. Đảo ngược các phần tử của mảng.
- i. Kiểm tra mảng có là mảng tăng dần không  
Ví dụ: Mảng gồm các số : 1, 2, 4, 6, 7 là mảng tăng dần  
Mảng gồm các số: 1, 2, 2, 6, 7 không là mảng tăng dần

Mảng gồm các số: 1, 2, 6, 4, 7 không là mảng tăng dần

- k. Chèn x vào mảng tăng dần để mảng vẫn là mảng tăng dần.
- l. Kiểm tra mảng có là mảng đối xứng không  
Ví dụ: Mảng gồm các số : 1, 2, 4, 2, 1 là mảng đối xứng  
Mảng gồm các số : 1, 2, 2, 1 là mảng đối xứng

**2. Tạo một mảng 1 chiều chứa n số**

nguyên Gợi ý: làm 2 cách

Cách 1: nhập giá trị phần tử mảng;

Cách 2: dùng lệnh rand() để sinh số nguyên ngẫu nhiên (rand() thuộc thư viện stdlib.h).

- a. Xuất một mảng 1 chiều chứa n số nguyên.
- b. Tính giá trị trung bình các phần tử trong mảng.
- c. Xuất ra danh sách các phần tử ở vị trí chẵn trong mảng.
- d. Đếm phần tử dương trong mảng.
- e. Đếm số lần phần tử x xuất hiện trong mảng.
- f. Tìm vị trí của giá trị x trong mảng, x nhập từ phím.
- g. Tìm phần tử lớn nhất/nhỏ nhất trong mảng .
- h. Kiểm tra mảng số nguyên có tồn tại giá trị 0 hay không? Nếu có thì trả về 1, ngược lại trả về 0 (*sử dụng kỹ thuật đặt cờ hiệu*).
- i. Tìm số chẵn lớn nhất trong mảng, nếu mảng không có số chẵn thì trả về -1.
- j. Đếm số nguyên tố trong mảng.

**7.4. Bài tập về nhà:**

- 2. Cho một mảng 1 chiều chứa các số thực. Hãy viết một chương trình cho phép người dùng chọn một trong các yêu cầu sau:
  - a. Tạo một mảng 1 chiều chứa n số thực (nhập từ bàn phím và sinh tự động ngẫu nhiên).
  - b. Xuất một mảng 1 chiều chứa n số thực.
  - c. Tính tổng các giá trị mảng.
  - d. Tính giá trị trung bình các phần tử trong mảng.
  - e. Tính tổng các giá trị dương trong mảng.
  - f. Đảo ngược các giá trị mảng.
  - g. Kiểm tra tính đối xứng của mảng. Nếu mảng không đối xứng thì trả về -1.
  - h. Viết hàm tìm giá trị phần tử xuất hiện nhiều nhất trong mảng, nếu có nhiều

phần tử có số lần xuất hiện bằng nhau thì xuất phần tử đầu tiên.

- i. Viết hàm kiểm tra mảng có phải là mảng tăng không.
3. Viết chương trình thực hiện các chức năng sau (mỗi chức năng viết một hàm riêng):
  - a. Nhập mảng gồm  $n > 0$  các số nguyên dương (có kiểm tra dữ liệu nhập).
  - b. Xuất mảng gồm  $n > 0$  các số nguyên dương.
  - c. In ra các số nguyên tố có trong mảng.
  - d. Tìm giá trị max của các số chẵn trong mảng.
  - e. Tìm ước số chung lớn nhất của từng cặp số trong mảng.

Ví dụ với mảng gồm 4 số: 2, 6, 4, 9 thì in ra:

$USCLN(2,6)=2$

$USCLN(2,4)=2$

$USCLN(2,9)=1$

$USCLN(6,4)=2$

$USCLN(6,9)=3$

$USCLN(4,9)=1$

- f. In ra biểu đồ nằm ngang của các số trong mảng.

Ví dụ với mảng gồm 4 số: 2, 6, 4, 9 thì in ra:

```
* *
* * * * *
* * *
* * * * *
```



## CHƯƠNG 8: MẢNG HAI CHIỀU

**Giới thiệu:** Cung cấp kiểu dữ liệu mảng 2 chiều cho người học. Cách khai báo và sử dụng mảng 2 chiều. Duyệt mảng 2 chiều và các thao tác trên mảng 2 chiều

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Khai báo được cấu trúc dữ liệu mảng 2 chiều.
- Phân tích được tình huống sử dụng mảng 2 chiều để lưu trữ dữ liệu.
- Cài đặt được các hàm nhập, xuất, xử lý dữ liệu trong các thao tác tính tổng, đếm giá trị phân tử.
- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

**Nội dung chính:**

### 8.1. Tóm tắt lý thuyết

#### 8.1.1. Định nghĩa:

Mảng hai chiều (còn gọi là ma trận) thực chất là mảng một chiều trong đó mỗi phần tử của mảng là một mảng một chiều, và được truy xuất bởi chỉ số dòng và chỉ số cột.

#### 8.1.2. Cú pháp:

**<tên kiểu dữ liệu> <tên mảng> [<số dòng>] [<số cột>];**

**Lưu ý:** Chỉ số cột và chỉ số dòng luôn luôn bắt đầu từ 0.

Ví dụ:

- ❑ Khai báo mảng 2 chiều a có 3 dòng 2 cột chứa số nguyên:

```
int a[3][2];
```

- ❑ Khai báo mảng 2 chiều ch có 50 dòng 100 cột chứa ký tự:

```
char ch[50][100];
```

#### 8.1.3. Truy xuất phần tử của mảng:

**Tên\_mảng [chỉ\_số\_dòng][ chỉ số cột]**

**Ví dụ:** Cho ma trận A như sau:

|             |   | Chỉ số cột |   |   |   |
|-------------|---|------------|---|---|---|
|             |   | 0          | 1 | 2 | 3 |
| Chỉ số dòng | 0 | 2          | 3 | 9 | 4 |
|             | 1 | 5          | 6 | 7 | 6 |
|             | 2 | 2          | 9 | 4 | 7 |

$A[0][0]=2; A[0][1]=3;$   
 $A[1][1]=6; A[1][3]=6;$

#### 8.1.4. Tham số mảng của một hàm

```

<Kiểu dữ liệu trả về>    <Tên hàm> (<kiểu dữ liệu ma trận>
<Tên ma trận>[<Số dòng tối đa>][<Số cột tối đa>], int m, int n)
{
    <Khởi tạo các biến>
    <Duyệt mảng để tính toán>
    <Trả về kết quả>
}

```

Trong đó, m là số dòng thực sự, n là số cột thực sự sử dụng lưu trữ trong ma trận.

**Qui ước:** Trong phần thực hành chúng ta thống nhất dùng **Ma trận** thay cho **Mảng 2 chiều**.

## 8.2. Bài tập mẫu hướng dẫn thực hành.

1. Viết chương trình nhập các giá trị cho ma trận chứa các số nguyên, sau đó xuất mảng vừa nhập (Lưu ý: Viết chương trình dạng hàm).

Yêu cầu:

- Viết 2 hàm con Nhập và Xuất ma trận chứa các số nguyên.
- Gọi thực hiện 2 hàm con trong hàm main. Hướng dẫn:

- Khai báo thêm các thư viện cơ bản cho chương trình.

```
#include <conio.h>
```

```
#include <stdio.h>
```

- Khai báo hằng số cho chương trình.

```
#define MAXROW 100
```

```
#define MAXCOL 100
```

- Viết các định nghĩa hàm cần thiết cho chương trình như sau:

```
//=====
```

```
void NhapMaTran_SoNguyen(int a[][MAXCOL], int &m, int &n)
```

```

{
    do
    {
        printf("Cho biet so dong
        cua mang: "); scanf_s("%d",
        &m);

    }while
    ile
    (m<
    =0)
    ; do
    {
        printf("Cho biet so cot cua mang: ");
        scanf_s("%d", &n);

    }while(n<=0);
    for(int i=0; i<m;
        i++)
        for(int
            j=0; j<n;
            j++)
        {
            printf("Gia tri phan tu a[%d][%d]=", i, j);
            scanf_s("%d", &a[i][j]);

        }
    }

    //=====
    void XuatMaTran_SoNguyen(int a[][MAXCOL], int m, int n)
    {
        for(int i=0; i<m; i++)
        {
            for(int j=0; j<n; j++)
                printf("%5d",
                    a[i][j]);

            printf("\n");
        }
    }
}

```

- Viết hàm main để thực thi chương trình.

```
void main()
{
    int A[MAXROW][MAXCOL];
    int M, N; NhapMaTran_SoNguyen(A, M, N);
    printf("\nNội dung của mảng là: \n");
    XuatMaTran_SoNguyen(A, M, N);
    _getch();
}
```

2. Cho ma trận chứa các số nguyên. Viết chương trình theo kiểu menu cho phép người dùng chọn thực hiện một trong các thao tác sau (Lưu ý: Viết chương trình dạng hàm):

- Nhập ma trận số nguyên bằng cách tạo giá trị ngẫu nhiên cho các phần tử của ma trận

- Xuất ma trận số nguyên
- Xuất các giá trị của ma trận ra màn hình.
- Tính và xuất tổng các phần tử ma trận.
- Tìm max/min của các phần tử ma trận.
- Sắp xếp các giá trị của ma trận theo thứ tự tăng dần.

Hướng dẫn:

- Khai báo thêm các thư viện cơ bản cho chương trình.

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<math.h>
```

- Khai báo hằng số cho chương trình.

```
#define MAXROW 100 //số dòng tối đa
#define MAXCOL 100 //số cột tối đa
```

- Viết các định nghĩa hàm cần thiết cho chương trình như sau:

```
//=====
void TaoMaTran_SoNguyen(int a[][MAXCOL], int &m, int &n)
{
    //SV tự làm
```

```

    }

    //=====
    =====

    void XuatMaTran_SoNguyen(int a[][MAXCOL], int m, int n)
    {
        //SV tự làm
    }

    //=====
    =====

    int TongMaTran(int a[][MAXCOL], int m, int n)
    {
        //SV tự làm
    }

    //=====
    =====

    int MaxMaTran(int a[][MAXCOL], int m, int n)
    {
        //SV tự làm
    }

    //=====
    =====

    int MinMaTran(int a[][MAXCOL], int m, int n)
    {
        //SV tự làm
    }

    //=====
    =====

    void HoanVi(int &x, int &y)
    {
        int temp=x; x=y; y=temp;
    }

```

```
//=====
=====

void SapXepMaTranTang(int a[][MAXCOL], int m, int n)
{
    int sopt=m*n;

    for(int i=0; i<sopt-1;
        i++) for(int j=i+1;
            j<sopt; j++)

        if(a[i/n][i%n]>a[j/n][j%n])
            HoanVi(a[i/n][i%n], a[j/n][j%n]);
}
```

- Viết hàm main để thực thi chương trình.

```
//=====
=====

void Menu()
{
    printf("\n*****\n");
    printf("\n*                MENU *\n");
    printf("\n*-----\n");
    printf("\n*");
    printf("\n* 1. Khoi tao ma tran *\n");
    printf("\n* 2. Xuat noi dung cua ma tran *\n");
    printf("\n* 3. Tong gia tri cac phan tu cua ma tran *\n");
    printf("\n* 4. Gia tri\nphan tu lon nhat, nho nhat *\n");
    printf("\n* 5. Sap xep\nma tran tang dan *\n");
    printf("\n* 0. Thoat chuong trinh *\n");
    printf("\n*****\n");
}

//=====

void main()
{
    // SV tự tổ chức các lệnh cho chương trình thực thi theo
    yêu cầu:

    cho người dùng lựa chọn câu/bài muốn thực thi
```

}

### 8.3. Bài tập ở lớp

Bài 1. Cho một ma trận chứa các số nguyên. Viết một chương trình (*dạng menu*) thực hiện các yêu cầu sau:

1. Tạo ngẫu nhiên giá trị cho các phần tử của ma trận.
2. Xuất giá trị các phần tử của ma trận ra màn hình.
3. Tìm vị trí của giá trị  $x$  trong ma trận,  $x$  nhập từ phím.
4. Tìm phần tử max của dòng  $k$  trong ma trận.
5. Đếm phần tử dương trong ma trận.
6. Đếm số lần phần tử  $x$  xuất hiện trong ma trận.
7. Kiểm tra ma trận có vuông không?
8. Kiểm tra ma trận có toàn là số chẵn không?
9. Tìm dòng có tổng lớn nhất.
10. Tìm số chẵn lớn nhất trong ma trận.

Bài 2. Cho một ma trận vuông chứa các số nguyên. Viết một chương trình (*dạng menu*) thực hiện các yêu cầu sau:

1. Nhập ma trận vuông, các giá trị người dung nhập từ phím
2. Nhập ma trận vuông , các giá trị tạo ngẫu nhiên
3. Xuất ma trận vuông
4. Tính tổng đường chéo chính
5. Đếm số phần tử chính phương trên đường chéo phụ
6. Kiểm tra ma trận dương không?
7. Kiểm tra ma trận có đường chéo chính toàn là số 1 không?
8. Kiểm tra ma trận đối xứng không?
9. Tìm vị trí phần tử có số lần xuất hiện nhiều nhất trong ma trận không?
10. Tìm số nguyên tố lớn nhất trong ma trận

### 8.4. Bài tập về nhà:

Cho một ma trận **vuông** chứa các số nguyên. Viết một chương trình (*dạng menu*) thực hiện các yêu cầu sau:

1. Xuất các phần tử trên đường chéo phụ
2. Xuất các phần tử thuộc tam giác trên.
3. Tìm cột có tổng lớn nhất trong ma trận.
4. Tìm dòng có tổng chẵn nhỏ nhất trong ma trận.
5. Tính tổng các phần tử trên đường chéo chính.
6. Tính tổng các phần tử chẵn thuộc tam giác trên.
7. Tính tổng các phần tử lẻ thuộc tam giác dưới.
8. Xuất các phần tử thuộc các đường biên của ma trận
9. Xuất các dòng tăng dần
10. Xuất các cột giảm dần
11. Chuyển đổi vị trí 2 dòng  $k$  và  $l$  trong ma trận.

12. Chuyển đổi vị trí 2 cột  $k$  và  $l$  trong ma trận.
13. Sắp xếp ma trận theo quy tắc các phần tử trên dòng được sắp tăng dần
14. Sắp xếp ma trận theo quy tắc tổng các dòng tăng dần
15. Đếm số phần tử cực đại trong ma trận. biết phần tử cực đại là phần tử lớn hơn các phần tử xung quanh nó.

## CHƯƠNG 9: CHUỖI

**Giới thiệu:** Cung cấp kiến thức về mảng 1 chiều cho người học. Cách khai báo và sử dụng mảng 1 chiều. Duyệt mảng một chiều và các thao tác trên mảng 1 chiều

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Trình bày được khái niệm chuỗi.
- Phân biệt được chuỗi và mảng các ký tự
- Khai báo chuỗi theo yêu cầu.
- Sử dụng các hàm thông dụng về chuỗi trong lập trình
- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

**Nội dung chính:**

### 9.1. Tóm tắt lý thuyết

#### 9.1.1. Khái niệm chuỗi

Trong ngôn ngữ lập trình C/C++, chuỗi ký tự được xem như là mảng 1 chiều chứa các ký tự, được dùng để lưu các chuỗi ký tự.

- Thư viện chứa các hàm hỗ trợ xử lý chuỗi: <string.h>
- Chuỗi luôn có ký tự kết thúc chuỗi là '\0' (null). Vì vậy độ dài chuỗi = kích thước mảng -1.
- Cú pháp khai báo chuỗi: char <tên biến chuỗi> [độ dài chuỗi];
- Ví dụ:

```
char hoTen[41]; // họ tên dài tối đa 40 ký tự
```

```
char diaChi[100]; // địa chỉ dài tối đa 99 ký tự
```

#### 9.1.2. Một số hàm xử lý chuỗi:

- Nhập chuỗi:
  - + Cách 1: dùng **scanf** với mã định dạng %s
  - + Cách 2: dùng **gets**
- Xuất chuỗi
  - + Cách 1: dùng **printf** với mã định dạng %s
  - + Cách 2: dùng **puts**
- Một số hàm xử lý khác:
  - + `size_t strlen(const char *s)`: Tìm độ dài của chuỗi.
  - + `char *strcpy(char *dest, const char *src)`: Copy chuỗi src sang chuỗi dest
  - + `int strcmp(const char *s1, const char*s2)`: So sánh hai chuỗi, hàm trả về >1 nếu chuỗi s1>s2, trả về 0 nếu 2 chuỗi bằng nhau, và trả về -1 nếu s1<s2. Hàm có phân biệt chữ hoa, chữ thường
  - + `int strcmpi(const char *s1, const char *s2)`: So sánh hai chuỗi, hàm trả

về >1 nếu chuỗi  $s_1 > s_2$ , trả về 0 nếu 2 chuỗi bằng nhau, và trả về -1 nếu  $s_1 < s_2$ . Hàm không phân biệt chữ hoa, chữ thường.

(SV tham khảo phần phụ lục A – giáo trình môn Nhập môn lập trình)

## 9.2. Bài tập mẫu hướng dẫn thực hành.

1. 1Viết chương trình nhập vào 2 chuỗi  $s_1, s_2$  bất kỳ từ bàn phím. Thực hiện các yêu cầu sau:

- Xuất ra màn hình 2 chuỗi  $s_1, s_2$ .
- Cho biết độ dài mỗi chuỗi (thực hiện 2 cách: dùng hàm `strlen` có sẵn của C/C++; không dùng hàm có sẵn..
- Xuất kết quả so sánh giữa 2 chuỗi (dùng 2 cách: có và không có phân biệt chữ hoa và chữ thường): “ $s_1$  lớn hơn  $s_2$ ”, “ $s_1$  bằng  $s_2$ ”, “ $s_2$  lớn hơn  $s_1$ ”.
- Chuyển chuỗi  $s_1$  thành chuỗi toàn chữ hoa/ thành chuỗi toàn chữ thường.

Hướng dẫn:

- SV tự viết các hàm và chương trình theo yêu cầu.
- Viết hàm main để thực thi chương trình.
- Một số hàm gợi ý:

**//Nhập chuỗi**

```
void nhapChuoi (char s[200])
```

```
{  
    printf ("Nhap chuoi: ");  
    scanf ("%s", &s);
```

```
    //Hoặc dùng gets
```

```
    //gets(s);
```

```
}
```

```
//-----
```

**//Xuất chuỗi**

```
void xuatChuoi (char s[200])
```

```
{  
    printf ("Chuoi : %s", s);
```

```
    // Hoặc dùng puts
```

```
    //puts(s);
```

```
}
```

```
//So sánh 2 chuỗi và có phân biệt chữ hoa/thường
void soSanhChuoiPhanBiethoaThuong (char s1[200], char
s2[200])
{
    int kq =
    strcmp(s1,s2);
    if(kq>0)
        printf("Chuoi %s lon hon %s", s1, s2);
    else
        if(kq<0)
            printf("Chuoi %s nho hon %s", s1, s2);
        else
            printf("Chuoi %s bang %s", s1, s2);
}

//hàm chuyển thành chữ thường
void tolower(char s[100])
{
    for(int i=0;i<strlen(s);i++)
        if(s[i] >='A'&& s[i] <='Z')
            s[i]=s[i]+32;
}
```

### 9.3. Bài tập ở lớp

- Viết các hàm chuẩn hóa chuỗi sau:
  - Xóa khoảng trắng đầu chuỗi.
  - Xóa khoảng trắng cuối chuỗi.
  - Xóa khoảng trắng dư giữa các từ trong chuỗi (chỉ để 1 khoảng trắng giữa các từ).
  - Đổi chuỗi thành chữ thường, riêng các chữ cái đầu mỗi từ thành chữ hoa.
- Viết các hàm thao tác trên chuỗi họ tên:
  - Đảo chuỗi họ tên.

Ví dụ :

Tên ban đầu:” Phan

Minh Anh” Tên sau

khi đảo :” Anh Phan

Minh”

b. Tạo địa chỉ email theo chuỗi họ tên bằng 3 kiểu. Ví dụ :

Tên :” Phan Minh Anh”

Địa chỉ email :[phanminhanh@yahoo.com](mailto:phanminhanh@yahoo.com) hay

[phan\\_minh\\_anh@yahoo.com](mailto:phan_minh_anh@yahoo.com) hay

[anhpm@yahoo.com](mailto:anhpm@yahoo.com)

3. Đếm số từ trong chuỗi.
4. Đảo ngược chuỗi
5. Xuất từng từ trong chuỗi, mỗi từ là 1 dòng (*tìm hiểu lệnh strtok*).
6. Xóa tất cả khoảng trắng trong chuỗi
7. Đếm số các ký tự đặc biệt (không phải ký tự và ký số) trong chuỗi.
8. Xóa tất cả các ký tự đặc biệt (không phải ký tự và ký số) trong chuỗi.

#### 9.4. Bài tập về nhà:

1. Nhập vào chuỗi họ tên,
  1. Xuất họ của chuỗi đó.
  2. Kiểm tra chuỗi trên có phải tên là “Lan”.
  3. Xuất từng thành phần của họ tên: họ, tên lót, tên
  4. Giả sử tên nhập vào đều có tên lót là “Thị” hoặc “Văn”. Hãy cho biết tên nhập vào là tên nam hay nữ.
2. Cho 2 chuỗi s1 và s2:
  1. Kiểm tra xem chuỗi s2 có là con của s1, hoặc ngược lại.
  2. Tìm các từ xuất hiện trong cả 2 chuỗi.
  3. Chèn s2 vào s1 tại vị trí k.

### CHƯƠNG 10: CON TRỎ

**Giới thiệu:** Cung cấp kiểu dữ liệu con trỏ cho người học. Cách khai báo và sử dụng biến con trỏ. Cấp phát và giải phóng vùng nhớ cho biến con trỏ. Mảng con trỏ và các hàm thao tác trên mảng con trỏ

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Khai báo biến kiểu con trỏ
- Vận dụng để cấp phát vùng nhớ cho biến con trỏ.
- Truy xuất đến ô nhớ do con trỏ chỉ đến.
- Viết các hàm xử lý trên mảng động (khai báo bởi biến con trỏ)
- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

### **Nội dung chính:**

## **10.1. Tóm tắt lý thuyết**

### 10.1.1 Khái niệm

Mỗi dữ liệu đều có một địa chỉ tương ứng dùng để chỉ vị trí của dữ liệu đó trên bộ nhớ. Con trỏ (pointer) là biến mà nội dung của biến đó là một địa chỉ bộ nhớ (memory address) dùng để chỉ đến một dữ liệu nào đó. Dùng con trỏ ta có thể truy cập biến thông qua địa chỉ của biến.

### 10.1.2. Khai báo biến con trỏ

**<Kiểu dữ liệu>\* <Tên biến con trỏ>;**

**<Kiểu dữ liệu> \*<Tên biến con trỏ>;**

#### **Ví dụ 6.1:**

**int \*x,\*y;**//x,y là con trỏ lưu địa chỉ của một số nguyên **float \*z;** //z là con trỏ lưu địa chỉ của một số thực **char \*s;** //s là con trỏ lưu địa chỉ của một số ký tự

### 10.1.3. Sử dụng biến con trỏ trong các biểu thức

Để sử dụng được biến con trỏ ta cần phân biệt 2 ký hiệu cũng chính là 2 toán tử sau:

**\***: Toán tử gián tiếp (*indirection operator*) dùng để tham khảo đến giá trị

**&**: Toán tử địa chỉ (*address operator*) dùng để tham khảo đến địa chỉ của

### 10.1.4. Cấp phát vùng nhớ dùng toán tử new

Cú pháp:

```
<Biến con trỏ> = new <Kiểu dữ liệu>;
```

#### 10.1.5. Giải phóng vùng nhớ dung toán tử delete

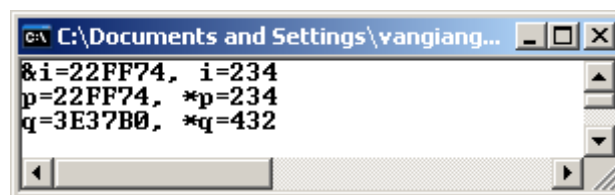
Cú pháp:

```
delete <biến con trỏ>;
```

**Ví dụ 6.6:** Xét chương trình in ra địa chỉ (theo dạng số hexadecimal) và nội dung của các biến như sau:

```
...
int main(){
    int i=234; int
    *p, *q; p = &i;
    q = new int;
    *q = 432;
    printf("&i=%X, i=%d\n", &i, i);
    printf("p=%X, *p=%d\n", p, *p);
    printf("q=%X, *q=%d\n", q, *q);
    delete q;
    getch();
    return 0;
}
```

Kết quả chạy chương trình (các giá trị địa chỉ mỗi lần chạy chương trình sẽ có thể là không giống nhau):



## 10.2. Bài tập mẫu hướng dẫn thực hành.

1. Viết chương trình nhập xuất mảng 1 chiều n số nguyên sử dụng con trỏ

```
int Menu() {
    int chon;
    printf("\n\n-----Menu-----\n");
    printf("\t0. Thoat\n");
    printf("\t1. Nhap mang\n");
    printf("\t2. Xuat mang\n");
}
```

```
        printf("\t3. Xuat mang nguoc\n");
        printf("\t4. Xuat chi so chan\n");
        printf    printf("MOI BAN CHON: "); scanf("%d",
&chon);
        return chon;
    }
void NhapMang(int *a, int &n){
    printf("Nhap n: ");
    scanf("%d", &n);
    a = new int[n];
    for(int i=0;i<n;i++){
        printf("Nhap a[%d] =  ", i);
        scanf("%d", (a+i));
    }
}
void XuatMang(int *a, int n){
    for(int i=0;i<n;i++)
        printf("%d\t", *(a+i));
    printf("\n");
}
void XuatMangNguoc(int *a, int n){
    for(int i=n-1;i>=0;i--)
        printf("%d\t", a[i]);
    printf("\n");
}
void XuatCSChan(int *a, int n){
    for(int i=0;i<n;i++)
        if(*(a+i)%2==0)
            printf("%d\t", *(a+i));
    printf("\n");
}

int main(){
    int *a,n,chon,kq,x,k;
    do{
        chon=Menu();
        if(chon==1){
            NhapMang(a, n);
        }
        else if(chon==2){
            printf("Cac pt trong mang:\n");

```

```
        XuatMang(a, n);
    }
    else if(chon==3){
        printf("Cac pt trong mang theo thu tu
nguoc:\n");
        XuatMangNguoc(a, n);
    }
    else if(chon==4){
        printf("Cac pt trong mang o chi so chan:\n");
        XuatCSChan(a, n);
    }
}while(chon!=0);
getch();
return 0;
}
```

### 10.3. Bài tập ở lớp

1. Viết chương trình dùng con trỏ thực hiện các chức năng sau (mỗi chức năng viết một hàm riêng):

- Dùng phương pháp cấp phát bộ nhớ động để nhập một dãy gồm n số nguyên. In ra màn hình:
- Tòan bộ dãy số theo thứ tự đã nhập
- Tòan bộ dãy số theo thứ tự ngược với thứ tự đã nhập
- Dãy con gồm các số tại các vị trí chỉ mục chẵn
- Dãy con gồm các số chẵn
- Đếm xem có bao nhiêu số chẵn
- Tính tổng các số chẵn của dãy số
- Tìm số lớn nhất trong dãy số
- Tìm số âm lớn nhất trong dãy số
- Xóa tất cả các số chẵn
- Chèn một số x vào vị trí k trong dãy số

2. Viết chương trình dùng con trỏ nhập vào một ma trận vuông cấp  $n > 2$ . Thực hiện:

- In ma trận ra màn hình
- Đếm số các phần tử có giá trị lẻ và tính tổng của chúng
- Tìm phần tử lớn nhất của từng dòng và nhỏ nhất của từng cột

### 10.4. Bài tập về nhà:

1. Viết chương trình dùng con trỏ nhập vào một danh sách nhiều họ tên sinh viên. Nhập vào một họ tên, tìm và cho biết có sinh viên nào trong danh sách trùng với họ tên đã nhập hay không. Yêu cầu dùng các hàm riêng cho từng chức năng.

2. Viết chương trình dùng con trỏ nhập vào một danh sách có nhiều tên sinh viên. Thay vì in các ký tự trong các chuỗi, in ra màn hình mã ASCII tương ứng với từng ký tự.

Ví dụ:

Nhập: **"Son", "Nam", "Tuan"**

Xuất: **"83 111 110", "78 97 109", "84 117 97 110"**

3. Viết chương trình dùng con trỏ cho phép người dùng nhập vào danh sách các sinh viên gồm họ tên, năm sinh, điểm. sau đó thực hiện các thao tác:

a. In ra toàn bộ danh sách đã nhập theo dạng bảng như sau:

| STT | Ho va ten      | Nam sinh | Diem |
|-----|----------------|----------|------|
| 1   | Tran Van Tuan  | 1992     | 7    |
| 2   | Nguyễn Thị Lan | 1993     | 8    |
| ... |                |          |      |

b. Đếm xem có bao nhiêu sinh viên có điểm nhỏ hơn 5

## CHƯƠNG 11: KIỂU CẤU TRÚC

**Giới thiệu:** Cung cấp kiểu dữ liệu cấu trúc gồm nhiều thành phần cho người học. Cách khai báo và sử dụng biến cấu trúc. Truy xuất tới từng thành phần của biến kiểu cấu trúc, sử dụng mảng cấu trúc và các hàm thao tác trên mảng cấu trúc

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Khai báo được kiểu dữ liệu cấu trúc
- Truy xuất các phần tử trên mảng cấu trúc.
- Tham khảo đến từng thành phần trong mảng cấu trúc
- Vận dụng để viết các hàm xử lý mảng cấu trúc
- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

**Nội dung chính:**

### 11.1. Tóm tắt lý thuyết

#### 11.1.1. Định nghĩa:

Cấu trúc là một kiểu dữ liệu bao gồm nhiều thành phần có thể thuộc nhiều kiểu dữ liệu khác nhau. Các thành phần được truy nhập thông qua một tên.

#### 11.1.2. Khai báo struct

Cách 1:

```
struct <Tên cấu trúc>
{
    <Tên kiểu dữ liệu> <Tên thành phần 1>;
    <Tên kiểu dữ liệu> <Tên thành phần 2>;
};
```

Ví dụ:

```
struct SinhVien
{
    char mssv[15];
    char hoten[50];
    int solanvang;
};
```

Cách 2:

```
typedef struct <Tên cấu trúc>
{
    <Tên kiểu dữ liệu> <Tên thành phần 1>;
    <Tên kiểu dữ liệu> <Tên thành phần 2>;
}<Tên cấu trúc mới>;
```

Ví dụ:

```
typedef struct SinhVien
{
    char mssv[15];
    char hoten[50];
    int solanvang;
}Student;
```

#### 11.1.3. Khai báo biến sử dụng cấu trúc

<Tên cấu trúc> <Tên biến>;

Ví dụ:

```
SinhVien x;
```

Hoặc:

```
Student y;
```

#### 11.1.4. Truy cập thành phần của cấu trúc:

*Tên\_biến\_cấu\_trúc.tên\_thành\_phần*

Ví dụ:

```
SinhVien x;
puts("Nhap ho ten:");
gets(x.hoten);
printf("Nhap so lan vang mat:");
scanf("%d",&x.solanvang);
```

#### 10.1.5. Khai báo mảng struct:

Khai báo mảng các cấu trúc hoàn toàn tương tự như khai báo mảng khác, chỉ có điều ở phần kiểu sẽ là tên một cấu trúc.

<Tên cấu trúc> <Tên mảng>[<kích thước mảng>;

Ví dụ:

```
struct SinhVien
{
    char mssv[50]; char hoten[50]; int solanvang;
};
SinhVien a[10];
```

#### 10.1.6. Truy cập thành phần của mảng cấu trúc:

Việc truy xuất đến một phần tử của mảng như a[2] ta sẽ thu được một biến có cấu trúc SinhVien và có thể lại tiếp tục truy xuất đến các thành phần của nó.

Ví dụ:

```
printf("\n Nhap ma so sinh vien:"); gets(a[2].mssv);
```

```
printf("\n Nhap ho ten:");
gets(a[2].hoten);
printf("\n Nhap so lan vang:");
scanf("%d", &a[2].solanvang);
```

## 11.2. Bài tập mẫu hướng dẫn thực hành.

### 1. Khai báo cấu trúc CongNhan lưu trữ thông tin:

- Ten
- NamSinh
- GioLam

Yêu cầu:

- Nhập mảng gồm n công nhân, xuất mảng công nhân vừa nhập
- Liệt kê tên công nhân nhận thưởng biết rằng công nhân nhận

thưởng là các công nhân có giờ làm lớn hơn 40 giờ.

Hướng dẫn:

- Khai báo thêm các thư viện cơ bản cho chương trình.

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include <math.h>
```

- Khai báo struct CongNhan cho chương trình.

```
struct CongNhan
```

```
{
```

```
    char ten[50]; int namsinh; int giolam;
```

```
};
```

- Viết các định nghĩa hàm cần thiết cho chương trình như sau:

```
void NhapMangCN(int &n, CongNhan a[])
```

```
{
```

```
    do
```

```
    {
```

```
        printf("Cho biet so cong nhan: ");
```

```
        scanf_s("%d", &n);
```

```
    }while(n<=0);
```

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

```
    {
```

```
        printf("Thong tin cong nhan thu %d la:\n", i+1);
```

```
        printf("Ten la:");
```

```
        fflush(stdin);
```

```
        gets_s(a[i].ten,100);
```

```
        printf("Nam Sinh:");
```

```
        scanf_s("%d", &a[i].namsinh);
```

```

        printf("Gio lam:");
        scanf_s("%d", &a[i].giolam);
    }
}
//=====
=
void XuatMangCN(CongNhan a[], int n) //co the xuat dang
bang
{
    for(int i=0; i<n; i++)
    {
        printf("\n=====
        ");
        printf("\nThong tin cong nhan thu %d la:",
        i+1); printf("\nTen la: %s",a[i].ten);
        printf("\nNam Sinh la %d:",a[i].namsinh);
        printf("\nGio lam %d:",a[i].giolam);

    }
}
//=====
==
void XuatCN(CongNhan pCN,int i)
{
    printf("\nThong tin cong nhan nhan thuong la:",
    i+1); printf("\nTen la: %s",pCN.ten);
    printf("\nNam Sinh la %d:",pCN.namsinh);
    printf("\nGio lam %d:",pCN.giolam);
}
//=====
==
void ThuongCN(CongNhan a[], int n)
{
    printf("\n=====");
    printf("\nDanh sach cong nhan nhan thuong:");
    for(int i=0; i<n; i++)
    {
        if(a[i].giolam>40)
        {
            XuatCN(a[i],i);
        }
    }
}

```

```

    }
    //=====
    ==

```

- Viết hàm main để thực thi chương trình.

```

void main()
{
    CongNhan A[1000]; // Mảng struct công nhân
    int N; // N công nhân biến kiểu nguyên
    NhapMangCN(N,A);
    ThuongCN(A,N);
    XuatMangCN(A,N);
    getch();
}

```

### 11.3. Bài tập ở lớp

1. Cho danh sách lưu thông tin của các mặt hàng, thông tin gồm:

- Mã mặt hàng (chuỗi 10 ký tự)
- Tên mặt hàng (chuỗi 50 ký tự)
- Số lượng mặt hàng (số nguyên)
- Đơn giá mặt hàng (số thực, đơn vị 1000 VNĐ)

Viết chương trình thực hiện các chức năng sau, mỗi chức năng viết một hàm riêng

- a. Nhập danh sách các mặt hàng, lưu ý khi nhập cần kiểm tra trùng mã mặt hàng.
- b. Xuất danh sách các mặt hàng theo dạng bảng, và tính thành tiền của từng mặt hàng.

| STT | Mã hàng | Tên hàng | Số lượng | Đơn giá | Thành tiền |
|-----|---------|----------|----------|---------|------------|
| 1   | TV      | Tivi     | 100      | 2000    | 200000     |
| 2.  | TL      | Tu lanh  | 200      | 3000    | 600000     |
| ... |         |          |          |         |            |

- c. Tìm một mặt hàng theo mã mặt hàng.
- d. Xuất danh sách các mặt hàng có số lượng lớn hơn hay bằng 100.
- e. Đếm xem có bao nhiêu mặt hàng có số lượng lớn hơn hay bằng 100.

- f. Tính giá trung bình của các mặt có giá >1.000.000 VNĐ
- g. Xuất các mặt hàng có giá lớn nhất.
- h. Xuất các mặt hàng có số lượng nhỏ nhất.
- l. Cập nhật giảm giá các mặt hàng 10%.
- m. Xóa các mặt hàng có số lượng = 0.

2. Cho danh sách lưu thông tin của các sinh viên, thông tin gồm:

- Mã sinh viên (chuỗi 10 ký tự)
- Tên sinh viên (chuỗi 50 ký tự)
- Năm sinh (số nguyên)
- Điểm (số thực)

Viết chương trình thực hiện các chức năng sau, mỗi chức năng viết một hàm riêng

- a. Nhập danh sách các sinh viên và lưu vào file.
- b. Đọc dữ liệu từ file và lưu vào mảng.
- c. Xuất danh sách các sinh viên (từ mảng) theo dạng bảng.
- d. Tìm một sinh viên theo mã sinh viên.
- e. Xuất danh sách các sinh viên có điểm <5.
- f. Đếm xem có bao nhiêu sinh viên có điểm <5.
- g. Tính điểm trung bình của các sinh viên có năm sinh 1992.
- k. Xuất các sinh viên có điểm lớn nhất.
- l. Xóa các sinh viên có điểm = 0.

#### **11.4. Bài tập về nhà:**

1. Tạo file txt lưu thông tin N nhân viên theo định dạng sau:

Dòng đầu tiên lưu số lượng nhân viên N

4xN dòng tiếp theo lưu Mã số nhân viên, Tên nhân viên, Hệ số lương, Chức vụ. Mỗi thông tin được nằm trên 1 dòng

Ví dụ:

3

001

Nguyen Hoang Oanh

2.34

Nhan vien

003

Nguyen Thanh Binh

3.0

Truong phong

002

Tran Bao Tien

4

Giam doc

Sau đó viết hàm thực hiện các chức năng sau:

- Nhập mảng gồm n nhân viên bằng cách đọc dữ liệu từ file.
    - Xuất mảng nhân viên.
  - In danh sách lương thực lãnh của nhân viên gồm các thông tin: stt, mã nhân viên, tên nhân viên, thực lãnh. Trong đó, thực lãnh = hệ số lương \* lương cơ bản + phụ cấp chức vụ, trong đó lương cơ bản là 1 hằng số có giá trị 1,210 VNĐ. Phụ cấp chức vụ: phụ cấp 500 VNĐ cho giám đốc và 300 VNĐ cho trưởng phòng.
    - Tính tổng lương của tất cả nhân viên
    - Tìm lương thực lãnh cao nhất
    - Tìm thông tin nhân viên có lương thực lãnh cao nhất.
2. Tạo file txt lưu thông tin N môn học theo định dạng sau:
- MonHoc gồm Mã số môn, Tên môn, Số tiết, Hình thức.
  - Sau đó viết hàm thực hiện các chức năng sau:
    - Nhập mảng gồm N môn học bằng cách đọc dữ liệu từ file.
    - Xuất mảng môn học.
    - In thông tin môn học có thông tin không đúng (môn học lý thuyết là bội của 15 và môn học thực hành là bội của 30)
      - In danh sách số tín chỉ theo từng môn học, biết rằng số tín chỉ = số tiết / 15 (đối với môn học lý thuyết) , và số tín chỉ = số tiết / 30 (đối với môn học thực hành). Ghi chú, không in môn học không đúng thông tin.
    - Hiệu chỉnh thông tin môn học cho phù hợp (chỉ cho phép hiệu chỉnh số tiết và hình thức).
    - Xóa môn học khi nhập vào tên môn học.
    - Thêm 1 môn học mới vào cuối danh sách môn học.

## CHƯƠNG 12: ÔN TẬP

**Giới thiệu:** Hệ thống lại nội dung kiến thức môn học. Viết chương trình tổng hợp các nội dung đã học của môn học vận dụng vào bài toán cụ thể

**Mục tiêu:** Sau khi học xong chương này người đọc có thể:

- Kỹ năng lập trình các bài toán cơ bản tốt bằng cách chia thành các hàm nhỏ.
- Trình bày các khái niệm mảng, mảng cấu trúc, các hàm xử lý trên mảng, mảng cấu trúc.
- Sử dụng các hàm về chuỗi để lập trình.
- Viết các hàm xử lý chuỗi
- Độc lập, trách nhiệm, cẩn thận trong lập trình, phối hợp làm việc theo nhóm

**Nội dung chính:**

### 1.1. Tóm tắt lý thuyết

1. Hàm đệ quy:
2. Chuỗi
3. Mảng 1 chiều
4. Mảng cấu trúc

### 12.2. Bài tập mẫu hướng dẫn thực hành.

**Câu 1:**

Cho dãy số nguyên  $F(n)$  được xác định như sau:

$$F(n) = F(n-2) + 5 \cdot F(n-1) + 3, \quad (n \geq 3). \quad \text{Với } F(1) = 1 \text{ và } F(2) = 3.$$

- a. Viết hàm đệ quy để tính số  $F(n)$  trên. Hàm có prototype như sau: `int F(int n);`
- b. Tìm hiểu viết hàm khử đệ quy (dùng vòng lặp) để tính số  $F(n)$  trên.
- c. Viết hàm main để chạy chương trình

Hướng dẫn:

```
int F(int n)
{
    if (n==1) return 1;
    if (n==2) return 3;
    return F(n-2)+5*F(n-1)+3;
}
```

**Câu 2:**

Viết hàm đếm có bao nhiêu từ trong chuỗi  $s$  có ký tự đầu từ viết chữ thường.

Ví dụ: cho chuỗi  $s = \text{"Truong Cao dang Ly Tu Trong thanh pho Ho Chi Minh"}$

Số từ có ký tự đầu viết thường: 3

Viết hàm main để chạy chương trình

Hướng dẫn: duyệt từ đầu tới cuối chuỗi. Kiểm tra điều kiện đầu từ nếu là ký tự thường thì tăng biến đếm lên 1

**Câu 3:**

Cho một mảng a gồm n số nguyên được nhập theo thứ tự từ  $a[0]$  đến  $a[n-1]$ .

Hãy viết các hàm thao tác trên mảng n số nguyên này như sau và hàm main để chạy chương trình.

Chú ý: sinh viên viết hàm tạo menu cho người sử dụng lựa chọn

a) Hàm xuất ra màn hình 2 dòng:

Dòng đầu gồm các phần tử có giá trị là số lẻ.

Dòng thứ hai là tổng của các phần tử có giá trị là số lẻ.

Ví dụ với mảng: 4      9      15      2      3      32      72      11.

Thì xuất ra:      9      15      3      11

Tong = 38

b) Hàm đếm trong mảng a có bao nhiêu phần tử có giá trị là ước của x (với x là tham số của hàm có kiểu số nguyên) .

c) Viết hàm kiểm tra mảng có chứa ít nhất một số nguyên tố hay không

d) Tạo menu và hàm main:

**Câu 4: (3 điểm).**

Thông tin về SinhVien gồm các thành phần được khai báo như sau:

```
struct SinhVien {  
    char MaSV[10]; //mã sinh viên  
    char Ten[30]; //tên sinh viên.  
    char Khoa[20]; //khoa quản lý sinh viên đang theo học  
    int Lop; //lớp học của sinh viên  
    float DiemTB; //điểm trung bình  
};
```

a) Viết hàm nhập xuất cho mảng n SinhVien trên

- b) Hàm đếm trong n sinh viên trên, có bao nhiêu sinh viên có điểm trung bình từ 7 tới 9.  
 c) Hàm xuất ra thông tin những sinh viên thuộc khoa “CNTT” Thông tin xuất như sau:

| STT | MaSV | TenSV | DiemTB | KetQua |
|-----|------|-------|--------|--------|
|-----|------|-------|--------|--------|

-----

-

|   |       |            |     |     |
|---|-------|------------|-----|-----|
| 1 | SV001 | Le Huy     | 3.6 | Rot |
| 2 | SV004 | Pham Hoang | 7.8 | Dau |

Biết rằng cột KetQua = “Dau” nếu điểm tb  $\geq 4$ , ngược lại kết quả là “Rot”

- d) Viết hàm main để chạy chương trình, sinh viên tạo menu.

### 12.3. Bài tập ở lớp

Câu 1:

Cho dãy số nguyên K(n) được xác định như sau:

$K(n) = K(n-2) * K(n-1) + 1$ , ( $n \geq 3$ ). Với  $K(1) = 1$  và  $K(2) = 2$ .

- Viết hàm đệ quy để tính số K(n) trên. Hàm có prototype như sau: `int K(int n);`
- Tìm hiểu viết hàm khử đệ quy (dùng vòng lặp) để tính số K(n) trên.
- Viết hàm main để chạy chương trình

Câu 2:

Viết hàm xuất ra từ đầu trong chuỗi s.

Ví dụ: cho chuỗi s = “ Truong Cao dang Ly Tu Trong thanh pho Ho Chi Minh ”

Thì xuất ra màn hình: Tu dau: Truong

Viết hàm main để chạy chương trình

Câu 3:

Cho một mảng a gồm n số nguyên được nhập theo thứ tự từ  $a[0]$  đến  $a[n-1]$ .

Hãy viết các hàm thao tác trên mảng n số nguyên này như sau và hàm main để chạy chương trình.

Chú ý: sinh viên viết hàm tạo menu (có hàm nhập mảng) cho người sử dụng lựa chọn

- a) Hàm xuất ra màn hình 2 dòng:

Dòng đầu gồm các phần tử có giá trị là số nguyên tố.

Dòng thứ hai gồm các phần tử có giá trị không là số nguyên tố.

Ví dụ với mảng: 4      9      15      2      3      32      72      11.

Thì xuất ra:      2      3      11

            4      9      15      32      72

- b) Hàm đếm trong mảng a có bao nhiêu phần tử có giá trị từ x tới y (với x,y là tham số của hàm có kiểu số nguyên) .

- c) Viết hàm kiểm tra mảng có chứa toàn số chẵn không?

- d) Tạo menu và hàm main:

Câu 4:

Thông tin về NhanVien gồm các thành phần được khai báo như sau:

```
struct NhanVien {
char MaNV[10]; //mã nhân viên
char Ten[30]; //tên nhân viên.
char PB[20]; //Phòng ban của nhân viên
int NgayCong //số ngày công làm việc trong tháng
float Luong; //lương hàng tháng
};
```

- Viết hàm nhập xuất cho mảng n NhanVien trên
- Hàm tính tổng lương các nhân viên thuộc phòng ban “Ke Hoach”
- Hàm xuất ra thông tin những nhân viên thuộc phòng ban x (x là tham số) Thông tin xuất như sau:

| STT | MaNV  | TenNV      | TongLuong |
|-----|-------|------------|-----------|
| 1   | NV001 | Le Huy     | 15000000  |
| 2   | NV004 | Pham Hoang | 12543000  |

Biết rằng cột TongLuong = Luong + Phụ cấp.

Với Phụ cấp = 2000000 nếu ngày công  $\geq 25$  ngày, ngược lại phụ cấp = 1000000

- Viết hàm main để chạy chương trình, sinh viên tạo menu.

## 12.4. Bài tập về nhà:

### Câu 1:

Cho dãy số nguyên  $T(n)$  được xác định như sau:

$$T(n) = 2 * T(n-2) + 3 * T(n-1) + 1, (n \geq 3). \text{ Với } T(1) = 0 \text{ và } T(2) = 2.$$

- Viết hàm đệ quy để tính số  $T(n)$  trên. Hàm có prototype như sau: `int T(int n);`
- Tìm hiểu viết hàm khử đệ quy (dùng vòng lặp) để tính số  $T(n)$  trên.
- Viết hàm main để chạy chương trình

### Câu 2:

Viết hàm xuất ra từ cuối cùng trong chuỗi s.

Ví dụ: cho chuỗi s = “ Truong Cao dang Ly Tu Trong thanh pho Ho Chi Minh ”

Thì xuất ra màn hình: Tu cuoi: Minh

Viết hàm main để chạy chương trình

### Câu 3:

Cho một mảng a gồm n số nguyên được nhập theo thứ tự từ  $a[0]$  đến  $a[n-1]$ .

Hãy viết các hàm thao tác trên mảng n số nguyên này như sau và hàm main để chạy chương trình.

Chú ý: sinh viên viết hàm tạo menu (có hàm nhập mảng) cho người sử dụng lựa chọn

a) Hàm xuất ra màn hình 2 dòng:

Dòng đầu gồm các phần tử có giá trị là số không chia hết cho 3.

Dòng thứ hai gồm các phần tử có giá trị là số chia hết cho 3.

Ví dụ với mảng: 4      9      15      2      3      32      72      11.

Thì xuất ra:      4      2      32      11  
                     9      15      3      72

b) Hàm đếm trong mảng a có bao nhiêu phần tử có giá trị  $\leq x$  (với x là tham số của hàm có kiểu số nguyên) .

c) Viết hàm kiểm tra mảng có chứa ít nhất một chính phương hay không

d) Tạo menu và hàm main:

#### Câu 4:

Thông tin về SinhVien gồm các thành phần được khai báo như sau:

```
struct SinhVien {
    char MaSV[10]; //mã sinh viên
    char Ten[30]; //tên sinh viên.
    char Khoa[20]; //khoa quản lý sinh viên đang theo học
    int Lop; //lớp học của sinh viên
    float DiemTB; //điểm trung bình
};
```

a) Viết hàm nhập xuất cho mảng n SinhVien trên

b) Hàm đếm trong n sinh viên trên, có bao nhiêu sinh viên thuộc khoa “CK” có điểm tb $\geq$ 4.

c) Hàm xuất ra thông tin những sinh viên thuộc lớp “21C1-CMT1” Thông tin xuất như sau:

| STT | MaSV  | TenSV      | DiemTB | HocBong |
|-----|-------|------------|--------|---------|
| -   |       |            |        |         |
| 1   | SV001 | Le Huy     | 6      |         |
| 2   | SV004 | Pham Hoang | 8.2    | x       |

Biết rằng cột HocBong = “x” nếu điểm tb $\geq$ 8, ngược lại HocBong để chuỗi rỗng.

Viết hàm main để chạy chương trình, sinh viên tạo menu.

### **TÀI LIỆU THAM KHẢO**

- [1].Phạm Văn Ất, Kỹ thuật lập trình C, NXB Thống Kê 2020
- [2].Phạm Ngọc Cương, Giáo trình ngôn ngữ lập trình C/C++. Nxb Thông tin và Truyền thông 2019
- [3].Nguyễn Văn Thu - Nguyễn Anh Phương, Giáo trình Lập trình nâng cao C++, NXB Đại học Kinh tế quốc dân 2020
- [4]. Nguyễn Văn Giang, Giáo trình môn Kỹ thuật lập trình, Khoa CNTT trường CĐ Lý Tự Trọng.