



Nguyễn Việt Nam Sơn  
nvnamson@gmail.com

# KỸ THUẬT LẬP TRÌNH

CÂU LỆNH ĐIỀU KIỆN  
CÂU LỆNH RỄ NHÁNH

# NỘI DUNG BÀI HỌC

1. GIỚI THIỆU

2. CÂU LỆNH ĐIỀU KIỆN

3. CÂU LỆNH RẼ NHÁNH

4. CHIA SẺ KINH NGHIỆM KHI LẬP TRÌNH

5. BÀI TẬP ỨNG DỤNG TẠI LỚP

6. BÀI TẬP VỀ NHÀ

# 1. GIỚI THIỆU

- Câu lệnh điều kiện & câu lệnh rẽ nhánh là 2 câu lệnh thường xuyên gặp trong bất kỳ chương trình nào (dù nhỏ hay lớn).
- Câu lệnh điều kiện: Chương trình sẽ đi thực hiện công việc nếu như có điều kiện gì đó xảy ra ở trước đó (lưu ý là điều kiện phải xảy ra rồi sau đó mới thực hiện công việc)
- Câu lệnh rẽ nhánh: Có chức năng về cơ bản thì giống như câu lệnh điều kiện.
- Thông qua bài học này, các bạn sẽ nắm được những điều sau:
  1. Cách sử dụng 2 loại câu lệnh trên.
  2. So sánh ưu & nhược điểm của 2 loại câu lệnh trên.
  3. Trong trường hợp nào thì nên áp dụng câu lệnh nào ?

## 2. CÂU LỆNH ĐIỀU KIỆN

### DẪN DẮT VẤN ĐỀ

Ở ngoài đời ta cũng đã có những câu điều kiện như sau:

Vd:

Câu lệnh thiếu:

- Nếu trời mưa thì tôi nghỉ học.
- Nếu Ngọc Trinh hạ giá thì tôi sẽ đi ... :3

Câu lệnh đủ:

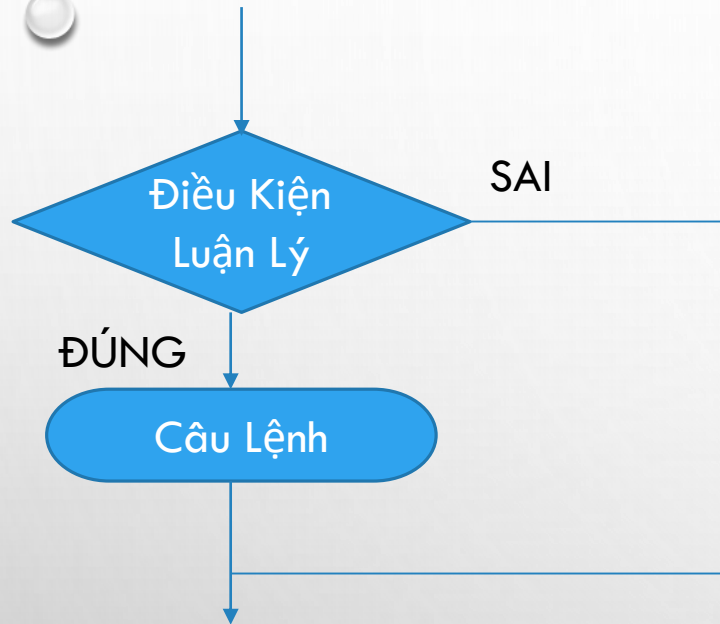
- Nếu trời mưa thì tôi nghỉ học, còn không thì tôi đi học.
- Nếu Ngọc Trinh hạ giá thì tôi sẽ đi, còn không thì tôi không đi.



Hành động chỉ xảy ra khi trước đó đã có điều kiện gì đó xảy ra

## 2. CÂU LỆNH ĐIỀU KIỆN

### CÂU LỆNH THIỂU



### **if(Điều Kiện Luận Lý)** **Câu Lệnh;**

- Nhìn vào hình vẽ ta hiểu quá trình như sau: Chương trình xét điều kiện xảy ra, nếu điều kiện đó là đúng thì sẽ đi thực hiện câu lệnh được cài đặt ở dưới. Nếu điều kiện đó là sai thì sẽ bỏ qua không thực hiện câu lệnh, chương trình sẽ tiếp tục chạy xuống dưới.
- Điều kiện luận lý: Cho kết quả đúng hoặc sai. Đúng sẽ là true hoặc hiểu là số khác 0. Sai sẽ là false hoặc hiểu là số 0.
- Câu lệnh: Có thể là “câu lệnh đơn” (Tức là chỉ có duy nhất 1 câu lệnh) hoặc là “câu lệnh phức” (Từ 2 câu lệnh trở lên). Nếu là câu lệnh phức thì phải đặt bên trong cặp ngoặc nhọn “{}”. Nếu là câu lệnh đơn thì không cần thiết.



# 2. CÂU LỆNH ĐIỀU KIỆN

## CÂU LỆNH THIỂU



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

```
int main()
{
    int a = 1, b = 2;

    if(a > b)
        printf("\na lon hon b");

    if(b > a)
    {
        a = a + 5;
        printf("\nb lon hon a");
    }

    getch();
    return 0;
}
```

Câu Lệnh Điều Kiện - Câu Lệnh Rẽ Nhánh



```
#include <iostream>
using namespace std;
```

```
int main()
{
    int a = 1, b = 2;

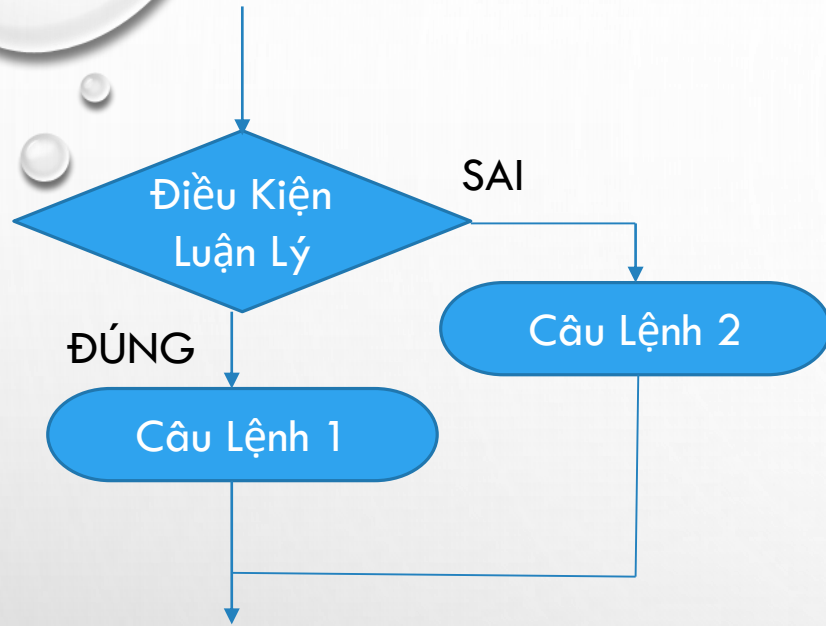
    if(a > b)
        cout << "\na lon hon b";

    if(b > a)
    {
        a = a + 5;
        cout << "\nb lon hon a";
    }

    system("pause");
    return 0;
}
```

## 2. CÂU LỆNH ĐIỀU KIỆN

### CÂU LỆNH ĐỦ



**if(Điều Kiện Luận Lý)**

**Câu Lệnh 1;**

**else**

**Câu Lệnh 2;**

- Nhìn vào hình vẽ ta hiểu quá trình như sau: Chương trình xét điều kiện xảy ra, nếu điều kiện đó là đúng thì sẽ đi thực hiện câu lệnh 1. Nếu điều kiện đó là sai thì sẽ đi thực hiện câu lệnh 2, xong rồi chương trình sẽ tiếp tục chạy xuống dưới.
- Điều kiện luận lý: Cho kết quả đúng hoặc sai. Đúng sẽ là true hoặc hiểu là số khác 0. Sai sẽ là false hoặc hiểu là số 0.
- Câu lệnh: Có thể là “câu lệnh đơn” (Tức là chỉ có duy nhất 1 câu lệnh) hoặc là “câu lệnh phức” (Từ 2 câu lệnh trở lên). Nếu là câu lệnh phức thì phải đặt bên trong cặp ngoặc nhọn “{}”. Nếu là câu lệnh đơn thì không cần thiết.

## 2. CÂU LỆNH ĐIỀU KIỆN

### CÂU LỆNH ĐỦ



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

```
int main()
{
    int a = 1, b = 2;

    if(a > b)
        printf("\na lon hon b");

    else
    {
        printf("\na nho hon hoac bang b");
        a = b;
    }

    getch();
    return 0;
}
```



```
#include <iostream>
using namespace std;
```

```
int main()
{
    int a = 1, b = 2;

    if(a > b)
        cout << "\na lon hon b";

    else
    {
        cout << "\na nho hon hoac bang b";
        a = b;
    }

    system("pause");
    return 0;
}
```



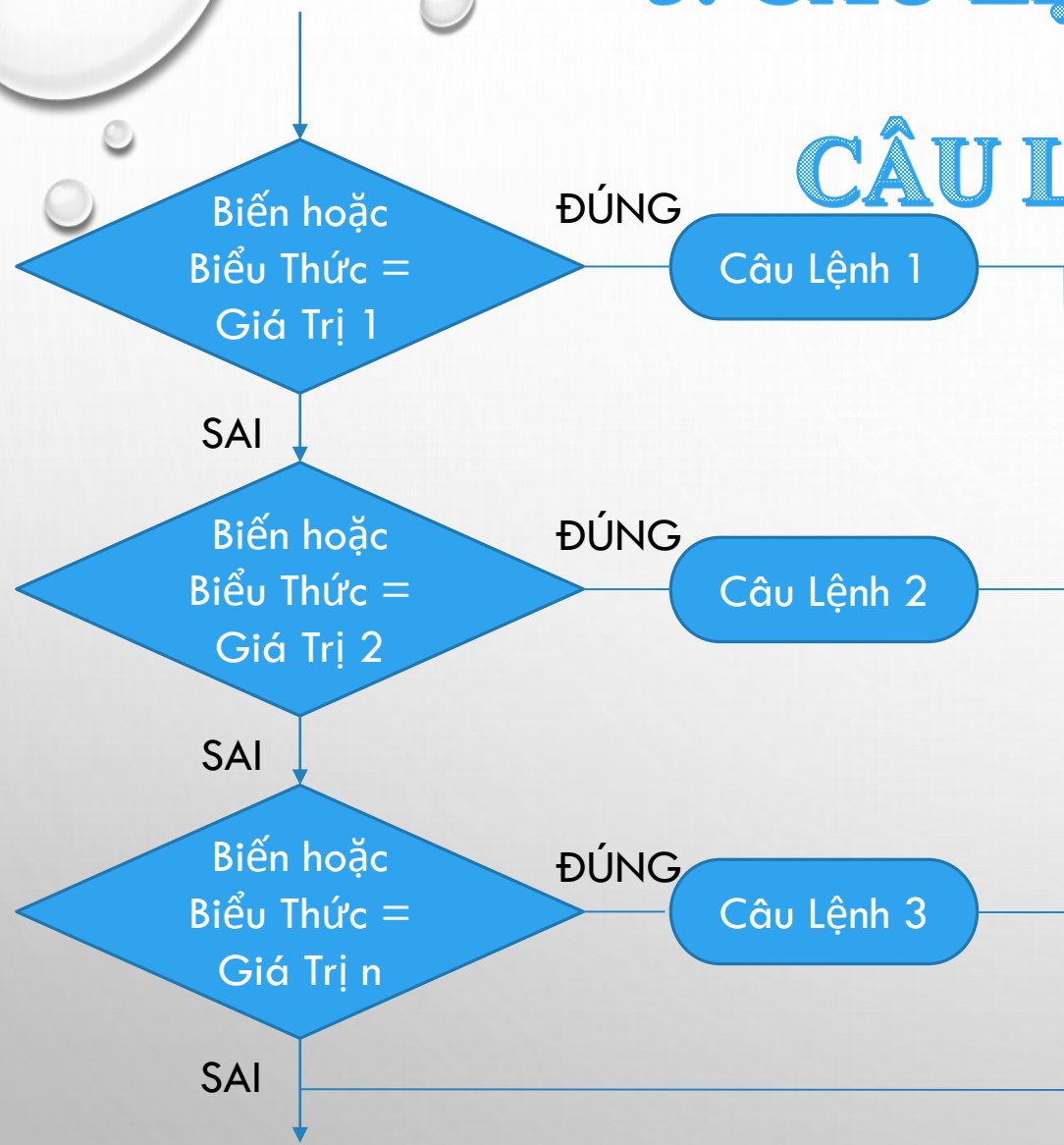
## 2. CÂU LỆNH ĐIỀU KIỆN

### CÓ THỂ BẠN KHÔNG BIẾT

1. Bản chất cả câu lệnh if hoặc cả câu lệnh if else đều chỉ là 1 câu lệnh đơn.
2. Câu lệnh else là phủ định trường hợp ngược lại của if ở trên nó và gần với nó nhất.
3. Các câu lệnh if else có thể lồng vào nhau với số lượng không giới hạn.
4. Đằng sau biểu thức điều kiện không được có “;”. Nếu như ta thêm “;” vào đằng sau if thì chương trình sẽ hiểu if đó bị “rỗng”.
5. Nên biết khi nào sử dụng toàn if, khi nào cần phải có else.

# 3. CÂU LỆNH Rẽ NHÁNH

## CÂU LỆNH THIỂU



Câu Lệnh Điều Kiện - Câu Lệnh Rẽ Nhánh

**switch**(Biến hoặc Biểu Thức)  
{

**case** <Giá trị 1>:  
    Câu Lệnh 1;  
    **break**;

**case** <Giá trị 2>:  
    Câu Lệnh 2;  
    **break**;

**case** <Giá trị n>:  
    Câu Lệnh n;  
    **break**;

}

- <Biến hoặc Biểu Thức>: Phải có 1 giá trị cụ thể và giá trị đó chỉ có thể là số nguyên hoặc là ký tự hoặc là chuỗi.
- <case: Giá Trị>: Kiểu dữ liệu của giá trị phải trùng với kiểu dữ liệu của <Biến hoặc Biểu Thức> và đặc biệt là các Giá Trị của các case không được trùng nhau.
- Câu Lệnh: Có thể là câu lệnh đơn hoặc câu lệnh phức { }

# 3. CÂU LỆNH RẼ NHÁNH

## CÂU LỆNH THIỂU

Viết chương trình nhập vào 1 số nguyên và kiểm tra nếu số này có giá trị nằm trong đoạn từ 0 tới 3 thì đọc nó lên (đọc thành chữ)



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

```
int main()
{
    int n;
    printf("\nNhap vao n = ");
    scanf("%d", &n);

    switch(n)
    {
        case 0: printf("\nKhong"); break;
        case 1: printf("\nMot"); break;
        case 2: printf("\nHai"); break;
        case 3: printf("\nBa"); break;
    }
    getch();
    return 0;
}
```



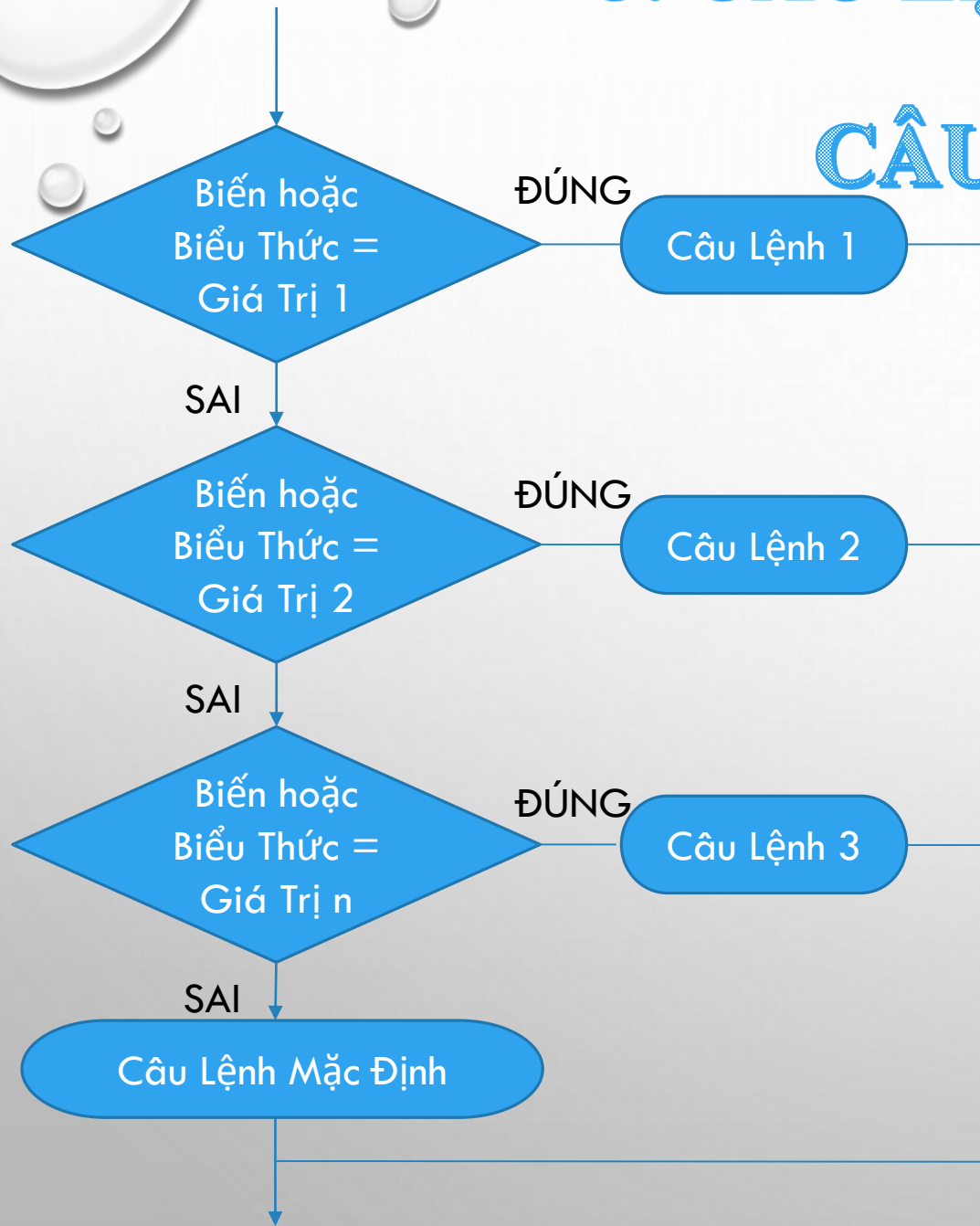
```
#include <iostream>
using namespace std;
```

```
int main()
{
    int n;
    cout << "\nNhap vao n = ";
    cin >> n;

    switch(n)
    {
        case 0: cout << "\nKhong"; break;
        case 1: cout << "\nMot"; break;
        case 2: cout << "\nHai"; break;
        case 3: cout << "\nBa"; break;
    }
    system("pause");
    return 0;
}
```

# 3. CÂU LỆNH Rẽ NHÁNH

## CÂU LỆNH ĐỦ



**switch**(Biến hoặc Biểu Thức)  
{

**case** <Giá trị 1>:  
    Câu Lệnh 1;  
    **break**;

**case** <Giá trị 2>:  
    Câu Lệnh 2;  
    **break**;

**case** <Giá trị n>:  
    Câu Lệnh n;  
    **break**;

**default**:  
    Câu Lệnh Mặc Định;

}

- <Biến hoặc Biểu Thức>:  
Phải có 1 giá trị cụ thể và giá trị đó chỉ có thể là số nguyên hoặc là ký tự hoặc là chuỗi.
- <case: Giá Trị>: Kiểu dữ liệu của giá trị phải trùng với kiểu dữ liệu của <Biến hoặc Biểu Thức> và đặc biệt là các Giá Trị của các case không được trùng nhau.
- Câu Lệnh: Có thể là câu lệnh đơn hoặc câu lệnh phức {}



# 3. CÂU LỆNH RẼ NHÁNH

## CÂU LỆNH ĐỦ

Viết chương trình nhập vào 1 số nguyên và kiểm tra nếu số này có giá trị nằm trong đoạn từ 0 tới 3 thì đọc nó lên (đọc thành chữ), nếu không thì báo là không đọc được.



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

```
int main()
{
    int n;
    printf("\nNhap vao n = ");
    scanf("%d", &n);

    switch(n)
    {
        case 0: printf("\nKhong"); break;
        case 1: printf("\nMot"); break;
        case 2: printf("\nHai"); break;
        case 3: printf("\nBa"); break;
        default: printf("\nKhong duoc duoc");
    }
    getch();
    return 0;
}
```



```
#include <iostream>
using namespace std;
```

```
int main()
{
    int n;
    cout << "\nNhap vao n = ";
    cin >> n;

    switch(n)
    {
        case 0: cout << "\nKhong"; break;
        case 1: cout << "\nMot"; break;
        case 2: cout << "\nHai"; break;
        case 3: cout << "\nBa"; break;
        default: cout << "\nKhong duoc duoc";
    }
    system("pause");
    return 0;
}
```



# 3. CÂU LỆNH RỄ NHÁNH


## CÓ THỂ BẠN KHÔNG BIẾT

1. Nếu như trong case có từ 2 câu lệnh trở lên (câu lệnh phức) thì ta phải để tất cả các câu lệnh nằm trong 1 cặp ngoặc nhọn { }. Nếu không trong một số trường hợp chương trình sẽ chạy không như ý muốn.
2. Bản chất switch case cũng là 1 câu lệnh đơn và nó có thể lồng vào bên trong 1 case khác, mức độ lồng không giới hạn.
3. Các giá trị trong switch chỉ có thể là kiểu số nguyên, ký tự hoặc chuỗi. Các giá trị trong case không được trùng nhau và phải có cùng kiểu dữ liệu với giá trị trong switch.
4. Khi giá trị truyền vào switch thì nó sẽ so sánh lần lượt với các giá trị trong các case tương ứng, nếu case nào có giá trị trùng thì nó sẽ thực hiện câu lệnh có trong case đó.
5. Sau khi thực hiện xong câu lệnh nằm trong case tương ứng thì switch sẽ kết thúc nếu gặp câu lệnh break hoặc nếu nó đã là case nằm ở cuối cùng.
6. Nếu trong case mà không có câu lệnh break thì nó sẽ tự động đi xuống case bên dưới và tiếp tục thực hiện. Điều này dễ gây ra lỗi ngoài ý muốn.

## 4. CHIA SẺ KINH NGHIỆM KHI LẬP TRÌNH

Nhận biết khi nào nên sử dụng câu lệnh điều kiện, khi nào nên sử dụng câu lệnh rẽ nhánh ?

- Khi ta cần so sánh “bằng” giữa 2 giá trị với nhau thì sử dụng cái nào cũng được. Còn các loại so sánh còn lại ( $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $!=$ ) thì chỉ có thể sử dụng câu lệnh điều kiện.
- Nếu giá trị so sánh “bằng” không thuộc 1 trong 3 kiểu dữ liệu quy định của câu lệnh rẽ nhánh (ký tự, chuỗi, số nguyên) thì không thể sử dụng được.
- **Vậy nếu như chỉ có thể sử dụng mỗi so sánh bằng giữa 2 giá trị với nhau (kiểu ký tự, số nguyên, chuỗi) vậy thì switch case hơn được if chỗ nào ???**



Khi chương trình được khởi chạy, nếu các giá trị trong các case được xếp “liên tục” nhau (vd: 1, 2, 3, 4, 5 ...) thì trình biên dịch sẽ tạo ra 1 bảng ánh xạ để ánh xạ công việc cần làm với giá trị tương ứng, nếu các giá trị này cạnh nhau, trình biên dịch sẽ đưa nó vào 1 mảng. Do tổ chức đặc biệt như thế nên lệnh sau biên dịch ra sẽ nhanh hơn dùng if nhiều (vì if mỗi lần phải biên dịch lại). Như thế này sẽ giúp chúng ta tối ưu hóa được mã nguồn.

# 5. BÀI TẬP ỨNG DỤNG TẠI LỚP

## Bài Tập Căn Bản

1. Nhập vào 2 số nguyên a, b. In ra màn hình kết quả số lớn nhất, nhỏ nhất.
2. Giải phương trình bậc 1 có dạng:  $ax + b = 0$ .
3. Viết chương trình nhập vào 1 số có 2 chữ số, in ra cách đọc của số này.  
Vd: 12 => Mười Hai
4. Viết chương trình nhập vào 1 số có 3 chữ số, in ra cách đọc của số này.  
Vd: 123 => Một Trăm Hai Mười Ba
5. Viết chương trình tính lương của nhân viên dựa theo thâm niên công tác (TNCT) như sau:  
Lương = hệ số \* lương căn bản, trong đó lương căn bản là 650000 đồng.
  - Nếu  $TNCT < 12$  tháng: hệ số = 1.92
  - Nếu  $12 \leq TNCT < 36$  tháng: hệ số = 2.34
  - Nếu  $36 \leq TNCT < 60$  tháng: hệ số = 3
  - Nếu  $TNCT \geq 60$  tháng: hệ số = 4.5

## 6. BÀI TẬP ỨNG DỤNG TẠI LỚP

### Bài Tập Nâng Cao

Một điểm KARAOKE tính tiền khách hàng theo công thức sau:

- Mỗi giờ trong 3 giờ đầu tiên tính 30 000 đồng/giờ,
- Mỗi giờ tiếp theo có đơn giá giảm 30% so với đơn giá trong 3 giờ đầu tiên.
- Ngoài ra nếu thời gian thuê phòng từ 8 – 17 giờ thì được giảm giá 10%.

Viết chương trình nhập vào giờ bắt đầu, giờ kết thúc và in ra số tiền khách hàng phải trả biết rằng  $8 \leq \text{giờ bắt đầu} < \text{giờ kết thúc} \leq 24$ .



## 6. BÀI TẬP VỀ NHÀ

### Bài Tập Căn Bản

1. Nhập vào 4 số nguyên a, b, c, d. In ra màn hình kết quả số lớn nhất, nhỏ nhất.
2. Giải phương trình bậc 2 có dạng:  $ax^2 + bx + c = 0$ .
3. Giải phương trình bậc 4 có dạng:  $ax^4 + bx^2 + c = 0$ .
4. Nhập vào tháng & năm. Hãy cho biết tháng đó trong năm đó có bao nhiêu ngày.
5. Nhập vào độ dài 3 cạnh của 1 tam giác. Kiểm tra xem 3 cạnh đó có tạo thành 1 tam giác hay không ? Và nếu có thì sẽ tạo thành tam giác gì (thường/cân/vuông/vuông cân/đều)  
Gợi ý: Điều kiện để tạo thành 1 tam giác đó là tổng của 2 cạnh bất kỳ luôn lớn hơn cạnh còn lại.
6. Nhập vào 1 ký tự là chữ cái. Nếu nhập sai thì báo thông báo đã nhập sai. Nếu đúng thì đi kiểm tra nếu đó đang là chữ cái thường thì biến nó thành chữ cái hoa, nếu nó là chữ cái hoa thì biến nó thành chữ cái thường.  
Gợi ý: Ký tự chữ cái thường và hoa có mã ASCII cách nhau 32 đơn vị. Vd: 'a' là 97, 'A' là 65



## 6. BÀI TẬP VỀ NHÀ

### Bài Tập Nâng Cao

1. Dạng của hệ phương trình bậc nhất 2 ẩn như sau:

$$ax + by = c$$

$$dx + ey = f$$

Trong đó:

+ a, b, c, d, e, f là các hệ số do người dung nhập vào

+ x & y là nghiệm cần tính

- Yêu cầu: Giải hệ phương trình trên.

2. Viết chương trình nhập vào 1 số nguyên dương giới hạn 9 chữ số. In ra cách đọc của số này.

Vd: 12345 => Mười hai ngàn ba trăm bốn mươi lăm

3. Nhập vào thông tin 1 ngày (ngày - tháng - năm). Kiểm tra ngày có hợp lệ hay không ? Nếu hợp lệ hãy tìm ra ngày kế tiếp (ngày - tháng - năm) & ngày trước đó (ngày - tháng - năm).

4. Kiểm tra 1 số nguyên dương x có phải là số chính phương hay không ?

Định nghĩa số chính phương: Là số mà kết quả khai căn bậc 2 của nó là 1 số nguyên. vd: 0, 1, 4, 9, 16, 25, 36 ...

## 6. BÀI TẬP VỀ NHÀ

### Bài Tập Nâng Cao

5. Một công ty trả lương cho nhân viên (theo tuần) như sau:

- Nhà quản lí (mã số 1) nhận một khoản lương cố định X (theo tuần).
- Công nhân làm theo giờ (mã số 2) nhận một khoản lương cơ bản Y đồng trong 40 giờ đầu trong tuần và 1.5 lần lương cơ bản cho mỗi giờ vượt mức quy định đó.
- Lương công nhân theo lợi nhuận (mã số 3) được nhận 500,000đ cộng thêm 7% trị giá doanh số Z mà công nhân ấy bán hàng ra trong tuần.
- Lương công nhân theo sản phẩm (mã số 4) nhận tiền dựa vào số N sản phẩm mà người đó đã làm ra trong tuần, với mỗi sản phẩm thì người này nhận được S đồng (mỗi người chỉ làm ra một loại sản phẩm nhất định).

Viết chương trình cho nhập vào mã số, tùy theo loại công nhân mà cho phép nhập vào những thông tin cần thiết cho công nhân đó để tính lương. Sau đó tính lương mà công ty phải trả cho nhân viên ấy trong tuần.