

BÀI TẬP THỰC HÀNH TUẦN 3**Mục đích**

- Giới thiệu các kiểu dữ liệu cơ bản.
- Các phép toán, phép dịch bit.

Công cụ

- Ngôn ngữ lập trình: C/C++
- Các ví dụ trong tài liệu này sử dụng Microsoft Visual C++ 2008

Các kiểu dữ liệu cơ bản

Bảng bên dưới thể hiện các kiểu dữ liệu cơ bản trong C/C++

STT	Tên KDL	Size		Chuỗi định dạng (stdio.h)
1	bool	1	0 or 1	Chỉ sử dụng cho C++
1	char	1 byte	-128 → 127 (hoặc 0 → 255 đối unsigned)	%c
2	short	2 byte	-32768 → 32767 (hoặc 0-65536 unsigned)	%d
3	int	Tùy môi trường, trên windows là 4 byte.		%d
4	long	4 byte		%ld
5	float	4 byte		%f
6	double	8 byte		%lf

Lab 01: Viết chương trình nhập vào bán kính đường tròn R, tính chu vi và diện tích của đường tròn.

```
//file lab1.cpp
```

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

```

#define PI 3.14159

int main()
{
    float r, cv, dt;
    printf(" r = ");
    scanf("%f", &r);
    cv = 2*PI*r;
    dt = PI*r*r;
    printf("Chu vi : %f", cv);
    printf("Dien tich : %f", dt);

    return 0;
}

```

Lab 02: Giải phương trình bậc nhất $ax + b = 0$, với a, b là số nguyên.

```

#include<stdio.h>

int main()
{
    int a, b;
    float x;
    printf(" Nhap a: ");
    scanf("%d", &a);
    printf(" Nhap b: ");
    scanf("%d", &b);

    if(a == 0)
    {
        if(b==0)
            printf("Phuong trinh dung voi moi x");
        else
            printf("Phuong trinh vo nghiem");
    }
    else
    {
        x = -b/a;
        printf("Phuong trinh co nghiem x=%f", x);
    }

    return 0;
}

```

Thực thi chương trình:

```
Nhap a:2
Nhap b:1
Phuong trinh co nghiệm x=0
```

Chương trình trên chạy sai do phép chia $x = -b/a$ là phép chia nguyên. Để có kết quả đúng, chúng ta phải thực hiện ép kiểu theo 1 trong 2 cách như sau:

```
x = -b/(float)a;
hoặc
x = -b*1.0/a; //ép kiểu tự nhiên.
```

Các phép toán trên kiểu dữ liệu cơ bản

Các phép toán trên kiểu dữ liệu cơ bản bao gồm các phép toán luận lý:

STT	Phép toán	Ý nghĩa	Ví dụ
1	==	Phép so sánh bằng	$a==b$, đúng nếu a có giá trị bằng b, ngược lại là sai.
2	>	So sánh lớn hơn	$a>b$, đúng nếu a lớn hơn b, ngược lại là sai.
3	<	So sánh nhỏ hơn	$a<b$, đúng nếu a nhỏ hơn b, ngược lại là sai.
4	!=	So sánh khác	$a!=b$, đúng nếu a khác b, ngược lại là sai.
5	>=	Lớn hơn hoặc bằng	$a>=b$
6	<=	Nhỏ hơn hoặc bằng	$a<=b$

Các phép toán số học: Các phép toán số học thực hiện cùng chức năng với phép toán trong toán học với một số lưu ý sau:

- Thứ tự thực hiện như với phép toán số học, nhân chia trước, cộng trừ sau, các phép toán trong ngoặc có độ ưu tiên cao hơn.
- Với phép chia : nếu các toán hạng là số nguyên thì phép chia là phép chia nguyên.
 - o Ví dụ: $c = 5/2$; sẽ cho kết quả là 2 thay vì 2.5
- Phép chia lấy phần dư là % với các toán hạng là số nguyên. Nếu có ít nhất một toán hạng không thuộc kiểu số nguyên thì phép chia % sẽ báo lỗi cú pháp.

Lab 03: Bài tập áp dụng:

Nhập vào một số tiền M đồng (bội số của 500). Đổi số tiền này ra các tờ giấy bạc 500.000đ, 200.000đ, 100.000đ, 50.000đ, 1000đ, 500đ. Với giả thiết ưu tiên cho tờ có mệnh giá lớn hơn, hãy in ra xem đổi được bao nhiêu tờ mỗi loại.

Hướng dẫn:

Vì ưu tiên cho tờ có mệnh giá lớn hơn nên chúng ta sẽ kiểm tra xem số tiền chúng ta đang có lần lượt có chia hết cho mệnh giá của tờ 500.000đ hay không, sau đó lấy phần dư còn lại để kiểm tra đối với tờ giấy bạc có mệnh giá 200.000đ và cứ như vậy cho đến tờ giấy bạc 500đ

```
#include<stdio.h>

int main()
{

    int m, k;

    printf("Nhap so tien(la boi so cua 500): ");
    scanf("%d", &m);

    if( (m/500000) > 0)
    {
        k = m/500000;
        printf("So to 500000 la %d\n", k);
        m = m%500000;
    }

    if( (m/200000) > 0)
    {
        k = m/200000;
        printf("So to 200000 la %d\n", k);
        m = m%200000;
    }

    if( (m/100000) > 0)
    {
        k = m/100000;
        printf("So to 100000 la %d\n", k);
        m = m%100000;
    }

    if( (m/50000) > 0)
    {
        k = m/50000;
        printf("So to 50000 la %d\n", k);
        m = m%50000;
    }

    if( (m/1000) > 0)
    {
        k = m/1000;
        printf("So to 1000 la %d\n", k);
        m = m%1000;
    }

    if( (m/500) > 0)
    {
        k = m/500;
        printf("So to 500 la %d", k);
    }
}
```

```
return 0;
}
```

Các phép toán trên bit (Bitwise operators): Các phép toán trên bit bao gồm AND(&), OR(|), XOR(^), NOT (~) và 2 phép dịch bit <<, >>.

Phép dịch bit trái (<<):

$$n = a \ll k \quad \text{tương đương với} \quad n = a * 2^k$$

Phép dịch bit phải (>>):

$$n = a \gg k \quad \text{tương đương với} \quad n = a / 2^k$$

Đối với các phép toán trên bit, một yêu cầu cần được chú ý là các toán hạng phải thuộc kiểu nguyên và cùng kích thước (size).

Bài tập cơ bản

- Viết chương trình nhập vào 3 số nguyên a, b, c. Cho biết số lớn nhất và nhỏ nhất trong 3 số trên.
- Một hình chữ nhật có chu vi là CV và có chiều dài bằng 1.5 lần chiều rộng. Viết chương trình nhập CV từ bàn phím, tính và in ra diện tích của hình chữ nhật.
- Viết chương trình nhập vào một số nguyên có 3 chữ số, cho biết có bao nhiêu chữ số khác nhau trong số nguyên vừa nhập vào:

Ví dụ 1:

Nhập: 345.

Xuất: có 3 chữ số khác nhau.

Ví dụ 2:

Nhập: 200.

Xuất: có 2 chữ số khác nhau.

Ví dụ 3:

Nhập: 111.

Xuất: có 1 chữ số trong số nguyên trên.

- Viết chương trình nhập vào 1 số nguyên (kiểu char), in ra màn hình bit thứ 3 từ trái sang của số nguyên trên.
- Viết chương trình nhập vào 2 số nguyên (kiểu int) a và b. Cho biết bit thứ 5 từ trái sang của 2 số a và b giống nhau hay khác nhau.

6. Một người bán rượu có các thùng chứa dung tích 100lít, 70lít, 50lít, 20lít, 10lít và 5lít. Viết chương trình giúp người bán rượu chia N lít rượu vào các thùng chứa trên với ưu tiên chứa đầy các thùng lớn trước rồi mới đến các thùng nhỏ.

Bài tập nâng cao

Yêu cầu sinh viên làm lại các bài tập trên với dữ liệu nhập vào từ tham số dòng lệnh, sinh viên tự đề nghị cách tổ chức tham số.