

Chương 3

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI C++

0. BÀI TẬP

- ♦ Bài 1: Viết chương trình nhập vào một phân số. Hãy cho biết phân số đó là phân số âm hay dương hay bằng không.
- ♦ Bài 2: Viết chương trình nhập vào một ngày. Tìm ngày kế tiếp và xuất kết quả.
- ♦ Bài 3: Viết chương trình nhập tọa độ hai điểm trong không gian. Tính khoảng cách giữa chúng và xuất kết quả.

0. BÀI TẬP

- ♦ Bài 4: Viết chương trình nhập vào một ngày. Tìm ngày hôm qua và xuất kết quả.
- ♦ Bài 5: Viết chương trình nhập vào 2 phân số. Tìm phân số lớn nhất và kết quả.
- ♦ Bài 6: Viết chương trình nhập vào 2 số phức. Tính tổng, hiệu, tích và xuất kết quả.
- ♦ Bài 7: Viết chương trình nhập tọa độ 3 đỉnh A,B,C của 1 tam giác trong mặt phẳng Oxy. Tính chu vi, diện tích và tìm tọa độ trọng tâm.

0. BÀI TẬP

- ♦ Bài 8: Viết chương trình nhập tọa tâm và bán kính của một đường tròn. Tính diện tích và chu vi của đường tròn.
- ♦ Bài 9: Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả.

HƯỚNG DẪN BÀI 1

♦ **Bài 1: Viết chương trình nhập vào một phân số. Hãy cho biết phân số đó là phân số âm hay dương hay bằng không.**

HƯỚNG DẪN BÀI 1

```
1. #include <stdio.h>
2. #include <conio.h>
3. class CPhanSo
4. {
5.     private:
6.         int tu;
7.         int mau;
8.     public:
9.         void Nhap();
10.        void Xuat();
11.        int XetDau();
12. };
```

HƯỚNG DẪN BÀI 1

```
11. void main()  
12. {  
13.     CPhanSo a;  
14.     a.Nhap();  
15.     int kq = a.XetDau();  
16.     switch(kq)  
17.     {  
18.         case 1: printf("");  
19.             break;  
20.         case -1: printf("");  
21.             break;  
22.         case 0: printf("");  
23.             break;  
24.     }  
25. }
```

HƯỚNG DẪN BÀI 1

♦ Định nghĩa hàm nhập phân số

```
1. void CPhanSo::Nhap()  
2. {  
3.     printf("Nhap tu: ");  
4.     scanf("%d",&tu);  
5.     printf("Nhap mau: ");  
6.     scanf("%d",&mau);  
7. }
```


HƯỚNG DẪN BÀI 1

♦ Định nghĩa hàm xuất phân số

```
1. void CPhanSo::Xuat()  
2. {  
3.     printf("\n Tu: %d", tu);  
4.     printf("\n Mau: %d", mau);  
5. }
```

HƯỚNG DẪN BÀI 1

```
1. int CPhanSo::XetDau()  
2. {  
3.     if(tu*mau>0)  
4.         return 1;  
5.     if(tu*mau<0)  
6.         return -1;  
7.     return 0;  
8. }
```

```
int kq = a.XetDau();
```

HƯỚNG DẪN BÀI 2

♦ **Bài 2: Viết chương trình nhập vào một ngày. Tìm ngày kế tiếp và xuất kết quả.**

HƯỚNG DẪN BÀI 2

```
11. #include <stdio.h>
12. #include <conio.h>
13. class CNgay
14. {
15.     private:
16.         int ng;
17.         int th;
18.         int nm;
19.     public:
20.         void Nhap();
21.         void Xuat();
22.         CNgay KeTiep();
23.         int ktNhuan();
24.         int sttTrongNam();
25.         long SoThuTu();
26.         void KhoiTao(int, int);
27.         void KhoiTao(long);
28.         CNgay(int, int, int);
29. }
```

HƯỚNG DẪN BÀI 2

```
1. void main()  
2. {  
3.     CNgay a;  
4.     a.Nhap();  
5.     CNgay kq = a.KeTiep();  
6.     printf("Ngay ban dau:");  
7.     a.Xuat();  
8.     printf("\nNgay ke tiep:");  
9.     kq.Xuat();  
10. }
```

HƯỚNG DẪN BÀI 2

♦ Định nghĩa hàm

```
1. void CNgay::Nhap()  
2. {  
3.     printf("Nhap ngay: ");  
4.     scanf("%d", &ng);  
5.     printf("Nhap thang: ");  
6.     scanf("%d", &th);  
7.     printf("Nhap nam: ");  
8.     scanf("%d", &nm);  
9. }
```

HƯỚNG DẪN BÀI 2

♦ Định nghĩa hàm

```
1. void CNgay::Xuat()  
2. {  
3.     printf("\nNgay:%d", ng);  
4.     printf("\nThang:%d", th);  
5.     printf("\nNam:%d", nm);  
6. }
```

HƯỚNG DẪN BÀI 2

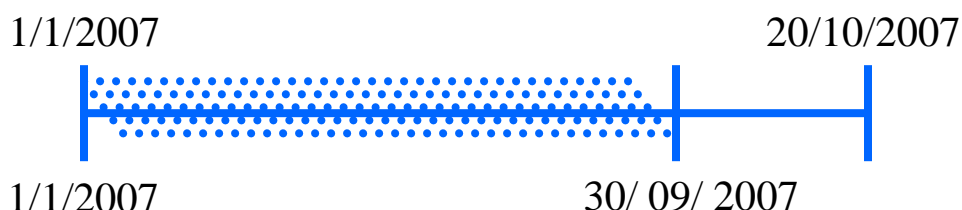
- ♦ Một năm được gọi là năm nhuận khi thoả một trong hai điều kiện.
 - Điều kiện 1: Năm chia hết cho 4 và không chia hết cho 100.
 - Điều kiện 2: Năm chia hết cho 400.
- ♦ Ví dụ 1: 1996 nhuận (điều kiện 1)
- ♦ Ví dụ 2: 2000 nhuận (điều kiện 2)
- ♦ Ví dụ 3: 1900 ko nhuận.
- ♦ Định nghĩa hàm

```
1. int CNgay::KiemTraNhuhan()  
2. {  
3.     if (nm%4==0 && nm%100!=0)  
4.         return 1;  
5.     if (nm%400==0)  
6.         return 1;  
7.     return 0;  
8. }
```


HƯỚNG DẪN BÀI 2

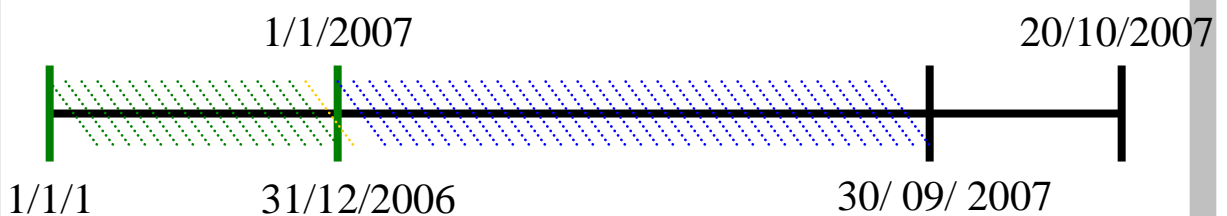
♦ Định nghĩa hàm

```
1. int CNgay::sttTrongNam()  
2. {  
3.     int ngaythang[12]={31,28,31,  
4.         30,31,30,31,31,30,31,30,31};  
5.     if (ktNhuan()==1)  
6.         ngaythang[1]=29;  
7.     int stt = 0;  
8.     for(int i=1;i<=th-1;i++)  
9.         stt=stt+ngaythang[i-1];  
10.    return (stt + ng);  
11. }
```



HƯỚNG DẪN BÀI 2

```
1. long CNgay::SoThuTu ()
2. {
3.     long stt = 0;
4.     for(int i=1;i<=nm-1;i++)
5.     {
6.         stt = stt + 365;
7.         CNgay temp(1,1,i);
8.         if(temp.ktnhuan()==1)
9.             stt = stt+1;
10.    }
11.    return (stt+sttTrongNam());
12. }
```



HƯỚNG DẪN BÀI 2

♦ Định nghĩa hàm

```
1. void CNgay::KhoiTao(int nam,int stt)
2. {
3.     int ngaythang[12]={31,28,31,
4.         30,31,30,31,31,30,31,30,31};
5.     CNgay temp(1,1,nam);
6.     if(temp.ktNhuan()==1)
7.         ngaythang[1] = 29;
8.     temp.th=1;
9.     while(stt-ngaythang[temp.th-1]>0)
10.    {
11.        stt=stt-ngaythang[temp.th-1];
12.        temp.th++;
13.    }
14.    temp.ng = stt;
15.    *this = temp;
```

HƯỚNG DẪN BÀI 2

♦ Định nghĩa hàm

```
1. void CNgay::KhoiTao(long stt)
2. {
3.     int nam = 1;
4.     int sn = 365;
5.     while(stt-sn>0)
6.     {
7.         stt = stt - sn;
8.         nam++;
9.         sn = 365;
10.        CNgay temp(1,1,nam);
11.        if( temp.ktnhuan()==1)
12.            sn = 366;
13.    }
14.    TimNgay(nam, (int)stt);
15. }
```

HƯỚNG DẪN BÀI 2

♦ Định nghĩa hàm

```
1. CNgay CNgay::KeTiep()  
2. {  
3.     long stt=SoThuTu();  
4.     stt=stt+1;  
5.     CNgay temp;  
6.     temp.KhoiTao(stt);  
7.     return temp;  
8. }
```

HƯỚNG DẪN BÀI 3

♦ **Bài 3: Viết chương trình nhập tọa độ hai điểm trong không gian. Tính khoảng cách giữa chúng và xuất kết quả.**

HƯỚNG DẪN BÀI 3

```
11. #include <stdio.h>
12. #include <conio.h>
13. class CDiemKhongGian
14. {
15.     private:
16.         float x;
17.         float y;
18.         float z;
19.     public:
20.         void Nhap();
21.         void Xuat();
22.         float KhoangCach
                (CDiemKhongGian);
23. } ;
```

HƯỚNG DẪN BÀI 3

```
11. void main()  
12. {  
13.     CDiemKhongGian A, B;  
14.     A.Nhap();  
15.     B.Nhap();  
16.     float kq=A.KhoangCach(B);  
17.     A.Xuat();  
18.     B.Xuat();  
19.     printf("\n...: %f", kq);  
20. }
```


HƯỚNG DẪN BÀI 3

♦ Định nghĩa hàm

```
11. void CDiemKhongGian::Nhap ()
12. {
13.     float temp;
14.     printf("Nhap x: ");
15.     scanf("%f", &temp);
16.     x = temp;
17.     printf("Nhap y: ");
18.     scanf("%f", &temp);
19.     y = temp;
20.     printf("Nhap z: ");
21.     scanf("%f", &temp);
22.     z = temp;
23. }
```

HƯỚNG DẪN BÀI 3

♦ Định nghĩa hàm

```
1. void CDiemKhongGian::Xuat()  
2. {  
3.     printf("\n x=%f", x);  
4.     printf("\n y=%f", y);  
5.     printf("\n z=%f", z);  
6. }
```

HƯỚNG DẪN BÀI 3

♦ Định nghĩa hàm

```
1. float CDiemKhongGian::KhoangCach  
    (CDiemKhongGian P)  
2. {  
3.     return sqrt( (x-P.x) * (x-P.x)  
        + (y-P.y) * (y-P.y)  
        + (z-P.z) * (z-P.z) );  
4. }
```

HƯỚNG DẪN BÀI 4

♦ **Bài 4: Viết chương trình nhập vào một ngày. Tìm ngày hôm qua và xuất kết quả.**

HƯỚNG DẪN BÀI 4

```
11. #include <stdio.h>
12. #include <conio.h>
13. class CNgay
14. {
15.     private:
16.         int ng;
17.         int th;
18.         int nm;
19.     public:
20.         void Nhap();
21.         void Xuat();
22.         CNgay HomQua();
23. };
```

HƯỚNG DẪN BÀI 4

```
1. void main()  
2. {  
3.     CNgay a;  
4.     a.Nhap();  
5.     CNgay kq = a.HomQua();  
6.     printf("Ngày ban dau:");  
7.     a.Xuat();  
8.     printf("\nNgày hom qua:");  
9.     kq.Xuat();  
10. }
```

HƯỚNG DẪN BÀI 5

♦ Định nghĩa hàm

```
1. void CNgay::Nhap()  
2. {  
3.     printf("Nhap ngay: ");  
4.     scanf("%d", &ng);  
5.     printf("Nhap thang: ");  
6.     scanf("%d", &th);  
7.     printf("Nhap nam: ");  
8.     scanf("%d", &nm);  
9. }
```

HƯỚNG DẪN BÀI 5

♦ Định nghĩa hàm

```
11. void CNgay::Xuat()  
12. {  
13.     printf("\nNgay:%d", ng);  
14.     printf("\nThang:%d", th);  
15.     printf("\nNam:%d", nm);  
16. }  
  
17. CNgay CNgay::HomQua()  
18. {  
19.     Em ko biết  
20. }
```


HƯỚNG DẪN BÀI 5

♦ **Bài 5: Viết chương trình nhập vào 2 phân số. Tìm phân số lớn nhất và kết quả.**

HƯỚNG DẪN BÀI 5

```
11. #include <stdio.h>
12. #include <conio.h>
13. class CPhanSo
14. {
15.     private:
16.         int tu;
17.         int mau;
18.     public:
19.         void Nhap();
20.         void Xuat();
21.         int SoSanh(CPhanSo);
22. };
```

HƯỚNG DẪN BÀI 5

```
1. void main()  
2. {  
3.     CPhanSo a,b;  
4.     printf("\n Nhap...:");  
5.     a.Nhap();  
6.     printf("\n Nhap...:");  
7.     b.Nhap();;  
8.     int kq = a.SoSanh(b);  
9.     printf("\nPhan so...:");  
10.    if(kq>=0)  
11.        a.Xuat();  
12.    else  
13.        b.Xuat();  
14. }
```

HƯỚNG DẪN BÀI 5

- ♦ Định nghĩa phương thức nhập phân số

```
1. void CPhanSO::Nhap()  
2. {  
3.     printf("Nhap tu: ");  
4.     scanf("%d",&tu);  
5.     printf("Nhap mau: ");  
6.     scanf("%d",&mau);  
7. }
```

HƯỚNG DẪN BÀI 5

- ♦ Định nghĩa phương thức xuất phân số

```
1. Void CPhanSo::Xuat()  
2. {  
3.     printf("\n Tu: %d", tu);  
4.     printf("\n Mau: %d", mau);  
5. }
```

HƯỚNG DẪN BÀI 5

```
11. int CPhanSo::SoSanh (CPhanSo x)
12. {
13.     float a=(float)tu/mau;
14.     float b=(float)x.tu/x.mau;
15.     if (a>b)
16.         return 1;
17.     if (a<b)
18.         return -1;
19.     return 0;
20. }
```

int kq = u.SoSanh(v) ;

HƯỚNG DẪN BÀI 6

♦ **Bài 6: Viết chương trình nhập vào 2 số phức. Tính tổng, hiệu, tích và xuất kết quả.**

HƯỚNG DẪN BÀI 6

```
11. #include <stdio.h>
12. #include <conio.h>
13. class CSoPhuc
14. {
15.     private:
16.         float thuc;
17.         float ao;
18.     public:
19.         void Nhap();
20.         void Xuat(S);
21.         CSoPhuc Tong(CSoPhuc);
22.         CSoPhuc Hieu(CSoPhuc);
23.         CSoPhuc Tich(CSoPhuc);
24. } ;
```


HƯỚNG DẪN BÀI 6

```
1. void main()  
2. {  
3.     CSoPhuc a,b,kq;  
4.     printf("\n Nhap...:");  
5.     a.Nhap();  
6.     printf("\n Nhap...:");  
7.     b.Nhap();  
8.     kq = a.Tong(b);  
9.     printf("\n Tong...:");  
10.    kq.Xuat();  
11.    kq = a.Hieu(b);  
12.    printf("\n Hieu...:");  
13.    kq.Xuat();  
14.    kq = a.Tich(b);  
15.    printf("\n Tich...:");  
16.    kq.Xuat();  
17. }
```

HƯỚNG DẪN BÀI 6

♦ Định nghĩa hàm

```
1. Void CSoPhuc::Nhap()  
2. {  
3.     float temp;  
4.     printf("Nhap thuc: ");  
5.     scanf("%f",&temp);  
6.     thuc = temp;  
7.     printf("Nhap ao: ");  
8.     scanf("%f",&temp);  
9.     ao = temp;  
10. }
```

HƯỚNG DẪN BÀI 6

♦ Định nghĩa hàm

```
1. void CSoPhuc::Xuat()  
2. {  
3.     printf("\nThuc=%f", thuc);  
4.     printf("\nAo = %f", ao);  
5. }
```

HƯỚNG DẪN BÀI 6

- ♦ Ví dụ: $\textcircled{3+5i}$ a
- ♦ $\textcircled{7+10i}$ b
- ♦ Tổng: $\textcircled{10+15i}$ temp

♦ Định nghĩa hàm

```

1. CSoPhuc CSoPhuc::Tong (CSoPhuc x)
2. {
3.     CSoPhuc temp;
4.     temp.thuc = thuc + x.thuc;
5.     temp.ao = ao + x.ao;
6.     return temp;
7. }
```

$kq = a.Tong(b);$

HƯỚNG DẪN BÀI 6

♦ Ví dụ: $3 + 5i$

♦
$$\frac{7 + 10i}{-4 - 5i}$$

♦ Hiệu: $-4 - 5i$

♦ Định nghĩa hàm

```
1. CSoPhuc CSoPhuc::Hieu(CSoPhuc x)
2. {
3.     CSoPhuc temp;
4.     temp.thuc = thuc - x.thuc;
5.     temp.ao = ao - x.ao;
6.     return temp;
7. }
```

HƯỚNG DẪN BÀI 6

- ♦ Ví dụ: $a + bi$
- ♦ $c + di$
- ♦ Tích $(a+bi)*(c+di)$
- ♦ $a(c+di)+bi(c+di)$
- ♦ $ac+adi+bci+bdi^2$
- ♦ $(ac-bd)+(ad+bc)i$

♦ Định nghĩa hàm

```

1. CSoPhuc CSoPhuc::Tich(CSoPhuc x)
2. {
3.     CSoPhuc temp;
4.     temp.thuc=thuc*x.thuc-ao*x.ao;
5.     temp.ao = thuc*x.ao+ao*c.tjic;
6.     return temp;
7. }
```

HƯỚNG DẪN BÀI 7

♦ Bài 7: Viết chương trình nhập tọa độ 3 đỉnh A, B, C của 1 tam giác trong mặt phẳng Oxy. Tính chu vi, diện tích và tìm tọa độ trọng tâm.

HƯỚNG DẪN BÀI 7

```
1. #include <stdio.h>
2. #include <conio.h>
3. class CDiem
4. {
5.     private:
6.         float x;
7.         float y;
8.     public:
9.         void Nhap();
10.        void Xuat();
11.        float KhoangCach(CDiem) ;
12. };
```


HƯỚNG DẪN BÀI 7

```
1. class CTamGiac
2. {
3.     private:
4.         CDiem A;
5.         CDiem B;
6.         CDiem C;
7.     public:
8.         void Nhap();
9.         void Xuat();
10.        float ChuVi();
11.        float DienTich();
12.        CDiem TrongTam();
13. };
```

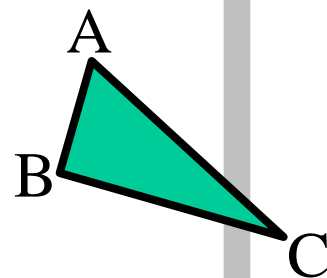
HƯỚNG DẪN BÀI 7

```
1. void main()  
2. {  
3.     CTamGiac tg;  
4.     tg.Nhap();  
5.     tg.Xuat();  
6.     float cv = tg.ChuVi();  
7.     printf("\n ...:%f",cv);  
8.     float dt=tg.DienTich();  
9.     printf("\n ...:%f",dt);  
10.    CDiem G = tg.TrongTam();  
11.    printf("\n...:");  
12.    G.Xuat();  
13. }
```

HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

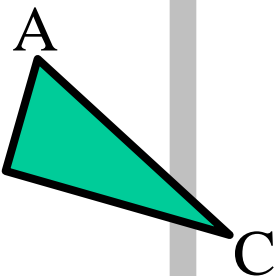
```
1. void CDiem::Nhap()
2. {
3.     float temp;
4.     printf("Nhap x: ");
5.     scanf("%f", &temp);
6.     x = temp;
7.     printf("Nhap y: ");
8.     scanf("%f", &temp);
9.     y = temp;
10. }
```



HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

```
1. void CTamGiac::Nhap ()
2. {
3.     printf("Nhap A: ");
4.     A.Nhap ();
5.     printf("Nhap B: "); B
6.     B.Nhap ();
7.     printf("Nhap C: ");
8.     C.Nhap ();
9. }
```



HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

```
11. void CDiem::Xuat()
```

```
12. {
```

```
13.     printf("\n x=%f", x);
```

```
14.     printf("\n y=%f", y);
```

```
15. }
```

```
16. void CTamGiac::Xuat()
```

```
17. {
```

```
18.     printf("\n A: ");
```

```
19.     A.Xuat();
```

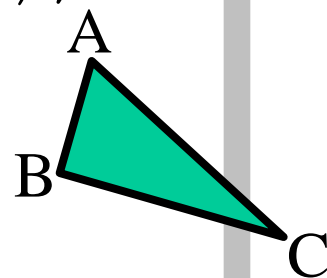
```
20.     printf("\n B: ");
```

```
21.     B.Xuat();
```

```
22.     printf("\n C: ");
```

```
23.     C.Xuat();
```

```
24. }
```

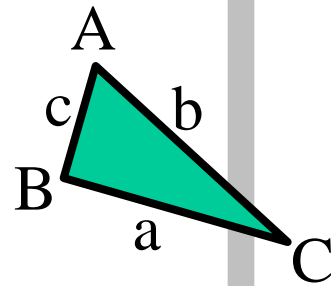


HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

```
1. float CDiem::KhoangCach (CDiem P)
2. {
3.     return sqrt ( (x-P.x) * (x-P.x)
4.                   + (y-P.y) * (y-P.y) ) ;
```

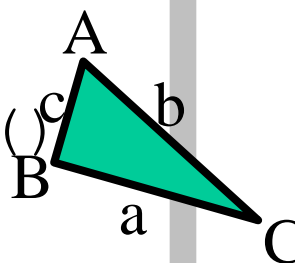
```
5. float CTamGiac::ChuVi ()
6. {
7.     float a=B.KhoangCach (C) ;
8.     float b=C.KhoangCach (A) ;
9.     float c=A.KhoangCach (B) ;
10.    return (a+b+c) ;
11. }
```



HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

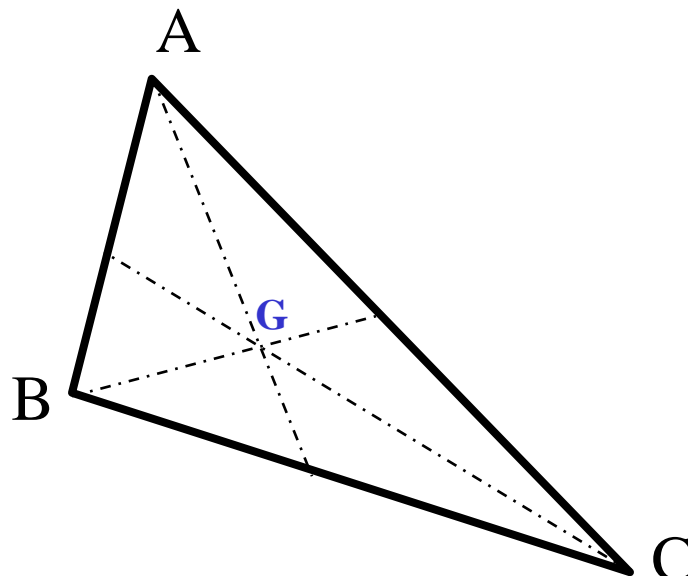
```
1. float CDiem::KhoangCach (CDiem P)
2. {
3.     return sqrt ( (x-P.x) * (x-P.x)
4.                   + (y-P.y) * (y-P.y) ) ;
5. }
6. float CTamGiac::DienTich ( )
7. {
8.     float a=B.KhoangCach (C) ;
9.     float b=C.KhoangCach (A) ;
10.    float c=A.KhoangCach (B) ;
11.    float p=(a+b+c) /2 ;
12.    return sqrt (p* (p-a) * (p-b) * (p-c) ) ;
13. }
```



HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

```
1. CDiem CTamGiac::TrongTam()  
2. {  
3.     CDiem temp;  
4.     temp.x = (A.x+B.x+C.x) / 3;  
5.     temp.y = (A.y+B.y+C.y) / 3;  
6.     return temp;  
7. }
```



HƯỚNG DẪN BÀI 8

♦ **Bài 8: Viết chương trình nhập tọa tâm và bán kính của một đường tròn. Tính diện tích và chu vi của đường tròn.**

HƯỚNG DẪN BÀI 8

```
11. #include <stdio.h>
12. #include <conio.h>
13. class CDiem
14. {
15.     public:
16.         float x;
17.         float y;
18.     public:
19.         void Nhap();
20.         void Xuat();
21. };
```

HƯỚNG DẪN BÀI 8

```
1. class CDuongTron
2. {
3.     private:
4.         CDiem I;
5.         float R;
6.     public:
7.         void Nhap();
8.         void Xuat();
9.         float ChuVi();
10.        float DienTich();
11. };
```

HƯỚNG DẪN BÀI 8

```
1. void main()  
2. {  
3.     CDuongTron c;  
4.     c.Nhap();  
5.     c.Xuat();  
6.     float cv = c.ChuVi();  
7.     printf("\n ...:%f", cv);  
8.     float dt = c.DienTich();  
9.     printf("\n ...:%f", dt);  
10. }
```

HƯỚNG DẪN BÀI 8

♦ Định nghĩa hàm

```
1. void CDiem::Nhap()
2. {
3.     float temp;
4.     printf("Nhap x: ");
5.     scanf("%f", &temp);
6.     x = temp;
7.     printf("Nhap y: ");
8.     scanf("%f", &temp);
9.     y = temp;
10. }
```

HƯỚNG DẪN BÀI 8

♦ Định nghĩa hàm

```
1. void CDuongTron::Nhap()
2. {
3.     float temp;
4.     printf("Nhap tam: ");
5.     I.Nhap();
6.     printf("Nhap ban kinh: ");
7.     scanf("%f", &temp);
8.     R = temp;
9. }
```

HƯỚNG DẪN BÀI 8

♦ Định nghĩa hàm

```
1. void CDiem::Xuat()
2. {
3.     printf("\n x=%f", x);
4.     printf("\n y=%f", y);
5. }
6. void CDuongTron::Xuat()
7. {
8.     printf("\n Tam: ");
9.     I.Xuat();
10.    printf("Ban kinh:%f",R);
11. }
```

HƯỚNG DẪN BÀI 8

♦ Định nghĩa hàm

```
1. float CDuongTron::ChuVi ()  
2. {  
3. |    return 2*3.14*R;  
4. }
```


HƯỚNG DẪN BÀI 8

♦ Định nghĩa hàm

```
1. float CDuongTron::DienTich ()  
2. {  
3.     |    return 3.14*R*R;  
4. }
```

HƯỚNG DẪN BÀI 9

- ♦ **Bài 9: Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả.**

HƯỚNG DẪN BÀI 9

```
1. #include <stdio.h>
2. #include <conio.h>
3. class CPhanSo
4. {
5.     private:
6.         int tu;
7.         int mau;
8.     public:
9.         void Nhap();
10.        void Xuat();
11.        CPhanSo Tong(CPhanSo);
12.        CPhanSo Hieu(CPhanSo);
13.        CPhanSo Tich(CPhanSo);
14.        CPhanSo Thuong(CPhanSo);
15. };;
```

HƯỚNG DẪN BÀI 9

```
1. void main()
2. {
3.     CPhanSo a,b,kq;
4.     printf("\n Nhap...:");
5.     a.Nhap();
6.     printf("\n Nhap...:");
7.     b.Nhap();
8.     kq = a.Tong(b);
9.     printf("\n Tong...:");
10.    kq.Xuat();
11.    kq = a.Hieu(b);
12.    printf("\n Hieu...:");
13.    kq.Xuat();
14.    kq = a.Tich(b);
15.    printf("\n Tich...:");
16.    kq.Xuat();
17. }
```

HƯỚNG DẪN BÀI 9

♦ Định nghĩa hàm nhập phân số

```
1. void CPhanSo::Nhap()  
2. {  
3.     printf("Nhap tu: ");  
4.     scanf("%d",&tu);  
5.     printf("Nhap mau: ");  
6.     scanf("%d",&mau);  
7. }
```

HƯỚNG DẪN BÀI 9

♦ Định nghĩa hàm xuất phân số

```
1. void CPhanSo::Xuat()  
2. {  
3.     printf("\n Tu: %d", tu);  
4.     printf("\n Mau: %d", mau);  
5. }
```

HƯỚNG DẪN BÀI 9

$$\begin{array}{c} \text{u} \\ \frac{a}{b} \end{array} + \begin{array}{c} \text{v} \\ \frac{c}{d} \end{array} = \begin{array}{c} \text{temp} \\ \frac{ad + bc}{bd} \end{array}$$

♦ Định nghĩa hàm

1. CPhanSo CPhanSo::Tong (CPhanSo x)

2. {

3. CPhanSo temp;

4. temp.tu= tu*x.mau + mau*x.tu;

5. temp.mau= mau * x.mau;

6. return temp;

7. }

kq = a.Tong(b);