

## Mục tiêu

Làm bài tập về quản lý bộ nhớ với con trỏ (pointer).

## Nội dung

Sửa Bài 4 của tuần 02

```
struct ThoiGian
{
    int gio, phut, giay;
};

ThoiGian* TaoThoiGian(int h, int m, int s)
{
    // ...
    ThoiGian *p = (ThoiGian*)malloc(sizeof(ThoiGian));
    if (p != NULL)
    {
        p->gio = abs(h);
        p->phut = abs(m);
        p->giay = abs(s);
    }
    return p;
}

void HopLeThoiGian(ThoiGian *pTG)
{
    // ...
    pTG->gio = abs(pTG->gio);
    pTG->phut = abs(pTG->phut);
    pTG->giay = abs(pTG->giay);

    pTG->phut += pTG->giay / 60;
    pTG->giay %= 60;
    pTG->gio += pTG->phut / 60;
    pTG->phut %= 60;
}

void PrintTG(ThoiGian *pTG)
{
    // ...
    printf("%2d:%2d:%2d", pTG->gio, pTG->phut, pTG->giay);
}

long TongGiayCuaTG(ThoiGian *pTG)
{
    return pTG->giay + pTG->phut * 60 + pTG->gio * 3600;
}

int SoSanhTG(ThoiGian *pTG1, ThoiGian *pTG2)
{
    long ts1 = TongGiayCuaTG(pTG1);
    long ts2 = TongGiayCuaTG(pTG2);
}
```

```
        if (ts1 > ts2)
        {
            return 1;
        }
        if (ts1 == ts2)
        {
            return 0;
        }
        return -1;
    }

    ThoiGian* KhoangCachThoiGian(ThoiGian *pTG1, ThoiGian *pTG2)
    {
        // ...
        HopLeThoiGian(pTG1);
        HopLeThoiGian(pTG2);
        ThoiGian *p, *q;
        int ss = SoSanhTG(pTG1, pTG2);
        if (ss > 0)
        {
            p = TaoThoiGian(pTG1->gio, pTG1->phut, pTG1->giay);
            q = TaoThoiGian(pTG2->gio, pTG2->phut, pTG2->giay);
        }
        else
        {
            if (ss == 0)
            {
                return TaoThoiGian(0, 0, 0);
            }
            else
            {
                p = TaoThoiGian(pTG2->gio, pTG2->phut, pTG2->giay);
                q = TaoThoiGian(pTG1->gio, pTG1->phut, pTG1->giay);
            }
        }
        if (p == NULL || q == NULL)
        {
            return p;
        }
        p->giay -= q->giay;
        if (p->giay < 0)
        {
            p->giay += 60;
            p->phut -= 1;
        }
        p->phut -= q->phut;
        if (p->phut < 0)
        {
            p->phut += 60;
            p->gio -= 1;
        }
        p->gio -= q->gio;
        return p;
    }
}
```

## Bài tập

### Bài 1

Viết chương trình, thực hiện các yêu cầu sau:

1. Nhập vào mảng các điểm (hoành độ x, tung độ y) trong mặt phẳng Oxy
2. Xuất mảng các điểm ra màn hình
3. Nhập vào tọa độ điểm A(x,y). Tìm điểm trong mảng cách xa điểm A nhất, nếu có nhiều điểm cùng xa A nhất thì chỉ trả về 1 điểm trong số đó.
4. Liệt kê các điểm có tọa độ dương trong mảng
5. Sắp xếp mảng các điểm tăng dần theo khoảng cách từ điểm đến gốc tọa độ O.

### Hướng dẫn

- Mở Visual Studio, tạo project với các thông số sau:
  - o Ngôn ngữ: Visual C++
  - o Tên project: Bai01
  - o Tên solution: MSSV\_Tuan03
  - o Loại project: Empty project
- Với project này, ta sẽ tạo 5 file:
  - o File Diem.h: chứa khai báo struct Diem và các hàm liên quan đến điểm
  - o File Diem.cpp: chứa cài đặt các hàm liên quan đến điểm
  - o File MangDiem.h: chứa khai báo các hàm liên quan đến mảng điểm
  - o File MangDiem.cpp: chứa cài đặt các hàm liên quan đến mảng điểm
  - o File main.cpp: chứa hàm main
- Với câu 1, nhập mảng điểm ta cần làm các việc sau:
  - o Viết hàm nhập mảng điểm
  - o Trong hàm nhập mảng điểm sẽ gọi hàm nhập điểm
  - o Do đó, cần khai báo struct Diem
- File Diem.h: khai báo struct Diem và hàm nhập thông tin 1 điểm:

```
#ifndef _DIEM_H_
#define _DIEM_H_
#include<stdio.h>
struct Diem
{
    float x, y;
};

void NhapDiem(Diem *d);

#endif
```

- File Diem.cpp: cài đặt hàm nhập thông tin 1 điểm

```
void NhapDiem(Diem *d)
{
    // B1: Nhập hoành độ x
    printf("Nhập x: ");
    scanf("%f", &d->x);
    // B2: Nhập tung độ y
    printf("Nhập y: ");
    scanf("%f", &d->y);
}
```

- File MangDiem.h: khai báo hàm nhập 1 mảng các điểm

```
#ifndef _MANGDIEM_H_
#define _MANGDIEM_H_
#include "Diem.h"
Diem* NhapMangDiem(int *n);
#endif
```

- File MangDiem.cpp: cài đặt hàm nhập 1 mảng các điểm

```
Diem* NhapMangDiem(int &n)
{
    // B1: Nhập số phần tử của mảng
    printf("Nhập n: ");
    scanf("%d", &n);
    // B2: Nhập giá trị cho từng điểm trong mảng
    Diem *a = new Diem[n];
    for(int i=0; i < n; i++)
    {
        printf("Nhập điểm thứ a[%d]", i);
        // B2.1: Gọi hàm NhapDiem
        NhapDiem(&a[i]);
    }
    return a;
}
```

- File main.cpp: chứa hàm main, khai báo mảng điểm và gọi hàm nhập mảng. Build #1, sửa lỗi, chạy thử chương trình

```
#include "Diem.h"
void main()
{
    Diem *a;
    int n;
```

## Thực hành Kỹ thuật lập trình

```
printf("Nhap mang cac diem, mang a\n");  
a = NhapMangDiem(n);  
  
}
```

- Với yêu cầu 2: xuất mảng các điểm ra màn hình
  - o Viết hàm xuất mảng các điểm. Hàm này sẽ gọi:
    - Hàm xuất thông tin 1 điểm ra màn hình
- File Diem.h: khai báo thêm hàm xuất điểm (ngay trước #endif)

```
void XuatDiem(Diem d);
```

- File Diem.cpp: cài đặt hàm xuất điểm

```
void XuatDiem(Diem d)  
{  
    // B1: Xuat x va y  
    printf("(%.3f,%.3f)", d.x, d.y);  
}
```

- File MangDiem.h: khai báo hàm xuất mảng điểm

```
void XuatMangDiem(Diem *a, int n);
```

- File MangDiem.cpp: cài đặt hàm xuất mảng điểm

```
void XuatMangDiem(Diem *a, int n)  
{  
    // B1: Duyệt từ đầu đến cuối mảng  
    for(int i=0; i<n; i++)  
    {  
        printf("Diem a[%d]", i);  
        // B1.1: Gọi hàm XuatDiem XuatDiem(a[i]);  
        printf("\n");  
    }  
}
```

- File main.cpp: gọi thêm hàm xuất mảng điểm, build #2, sửa lỗi, chạy thử

```
void main()  
{  
    int n;  
    printf("Nhap mang cac diem, mang a\n");  
    Diem *a = NhapMangDiem(&n);  
    printf("Mang a vua nhap\n");  
    XuatMangDiem(a, n);  
}
```

## Thực hành Kỹ thuật lập trình

```
}
```

- Với yêu cầu 3, ta viết hàm tìm phần tử trong mảng xa A nhất:
  - o Hàm này sẽ gọi hàm tính khoảng cách từ a[i] đến A để xem a[i] nào xa nhất
- Do đó, file Diem.h: khai báo hàm tính khoảng cách 2 điểm

```
float TinhKhoangCach(Diem a, Diem b);
```

- File Diem.cpp: cài đặt hàm tính khoảng cách

```
float TinhKhoangCach(Diem a, Diem b)
{
    return sqrt((b.x-a.x)*(b.x-a.x)
                + (b.y-a.y)*(b.y-a.y));
}
```

- File MangDiem.h: khai báo hàm tìm điểm cách xa nhất  
Diem TimXaNhat(Diem \*a, int n, Diem A);
- File MangDiem.cpp: cài đặt hàm tìm điểm cách xa nhất

```
Diem TimXaNhat(Diem *a, int n, Diem A)
{
    // B1: Giả sử a[0] là điểm xa A nhất
    Diem kq=a[0];
    // B2: Tính khoảng cách từ a[0] đến A
    float kc=TinhKhoangCach(a[0],A);
    // B3: Duyệt từ đầu đến cuối mảng
    for(int i=1;i<n;i++)
    {
        // B3.1: Tính khoảng cách từ a[i] đến A
        float kc2=TinhKhoangCach(a[i],A);
        // B3.2: Nếu kc2>kc
        // cập nhật kq và kc
        if(kc2>kc)
        {
            kq=a[i];
            kc=kc2;
        }
    }

    return kq;
}
```

```
}
```

- File main.cpp: gọi hàm, build #3, sửa lỗi, chạy thử

```
Diem A;  
printf("Nhap diem A\n");  
NhapDiem(&A);  
Diem x = TimXaNhat(a,n,A);  
printf("Diem trong mang xa A nhat la: ");  
XuatDiem(x);  
printf("\n");
```

- Gợi ý cho câu 4:

```
void LietKeToaDoDuong(Diem *a, int n)  
{  
  
    // B1: Duyệt từ đầu đến cuối mảng  
  
    // B1.1: Nếu a[i].x>0 và a[i].y>0  
  
    // B1.1.1: Gọi hàm XuatDiem a[i]  
  
}
```

- Gợi ý cho câu 5:

```
void SapTangTheoKhoangCach(Diem *a, int n)  
{  
    // B1: Khai báo điểm O  
    // ,gan O.x: 0  
    // ,gan O.y: 0  
    // B2: i duyệt từ 0 đến n-2  
    // B2.1: j duyệt từ i+1 đến n-1  
    // B2.1.1: Nếu khoảng cách (a[i], O)  
    // > khoảng cách (a[j], O)  
  
    // B2.1.1.1: Hoán vị a[i] và a[j]  
  
}
```

## Bài 2

Viết chương trình thực hiện các chức năng sau:

1. Nhập vào mảng các tam giác

2. Xuất mảng các tam giác ra màn hình
3. Tìm tam giác có chu vi lớn nhất
4. Tính tổng diện tích các tam giác
5. Đếm các tam giác có diện tích lớn hơn diện tích trung bình trong mảng
6. Sắp xếp mảng các tam giác giảm dần theo chu vi

### Hướng dẫn

- Để làm được bài này, chúng ta cần 7 file:
  - o File Diem.h: copy từ bài 1 qua xài (ai rảnh thì code lại cũng được)
  - o File Diem.cpp: copy từ bài 1 qua xài (ai rảnh thì code lại cũng được)
  - o File TamGiac.h: khai báo struct TamGiac và các hàm liên quan đến tam giác

```
#ifndef _TAMGIAC_H_
#define _TAMGIAC_H_
#include "Diem.h"
struct TamGiac
{
    Diem *A, *B, *C;
};
#endif
```

- o File TamGiac.cpp: cài đặt các hàm liên quan đến tam giác
- o File MangTamGiac.h: khai báo các hàm liên quan đến mảng tam giác

```
#ifndef _MANGTG_H_
#define _MANGTG_H_
#include "TamGiac.h"
#endif
```

- o File MangTamGiac.cpp: cài đặt các hàm liên quan đến mảng tam giác
  - o File main.cpp: chứa hàm main
- Yêu cầu 1: hàm nhập mảng các tam giác
    - o Gọi hàm nhập 1 tam giác
      - Gọi hàm nhập điểm 3 lần để nhập giá trị cho đỉnh A, B và C

```
void NhapTamGiac(TamGiac *t)
{
    printf("Nhap dinh A\n");
    t->A = new Diem;
    NhapDiem(t->A);
    // Lam tuong tu cho B va C
```



```
}  
  
TamGiac* NhapMangTamGiac(int &n)  
{  
    // B1: Nhap so phan tu, n  
    // B2: Cap phat vung nho cho mang  
    // B3: Duyet tu dau den cuoi mang  
    for(int i=0;i<n;i++)  
    {  
        printf("Nhap tam giac a[%d]\n",i);  
        // B3.1: Goi ham NhapTamGiac a[i]  
        NhapTamGiac(&a[i]);  
    }  
    return a;  
}
```

- Yêu cầu 2: hàm xuất mảng các tam giác
  - o Gọi hàm xuất 1 tam giác
    - Gọi hàm xuất điểm 3 lần để xuất giá trị cho đỉnh A, B, C

```
void XuatTamGiac(TamGiac t)  
{  
    printf("Dinh A: ");  
    XuatDiem(t.A);  
    printf("\n");  
    printf("Dinh B: ");  
    XuatDiem(t.B);  
    printf("\n");  
    printf("Dinh C: ");  
    XuatDiem(t.C);  
    printf("\n");  
}
```

```
void XuatMangTamGiac(TamGiac *a, int n)
{
    for(int i=0;i<n;i++)
    {
        printf("Tam giac a[%d]: ");
        // Goi ham XuatTamGiac a[i]
        printf("\n");
    }
}
```

- Yêu cầu 3: hàm tìm tam giác có chu vi lớn nhất
  - o Gọi hàm tính chu vi 1 tam giác
    - Gọi hàm tính khoảng cách 3 lần để tính khoảng cách 3 cạnh AB, BC, CA

```
float TinhChuVi(TamGiac t)
{
    float AB=TinhKhoangCach(t.A, t.B);
    float BC=TinhKhoangCach(t.B, t.C);
    float CA=TinhKhoangCach(t.C, t.A);
    return AB+BC+CA;
}

TamGiac TimChuViNhoNhat(TamGiac *a, int n)
{
    // B1: Gia su kq = a[0] la tam giac co chu vi nho nhat
    // B2: Duyet tu dau den cuoi mang
    // B2.1: Neu chu vi (a[i]) > chu vi (kq)
    // Cap nhat kq=[i]
}
```

## Thực hành Kỹ thuật lập trình

- Yêu cầu 4: hàm tính tổng diện tích của các tam giác
  - o Gọi hàm tính diện tích 1 tam giác
    - Gọi hàm tính khoảng cách 3 lần để tính khoảng cách 3 cạnh AB, BC, CA

```
float TinhDienTich(TamGiac t)
{
    float AB=TinhKhoangCach(t.A, t.B);
    float BC=TinhKhoangCach(t.B, t.C);
    float CA=TinhKhoangCach(t.C, t.A);
    // Tinh nua chu vi
    float p= (AB+BC+CA) /2;
    return sqrt(p*(p-AB)*(p-BC)*(p-CA));
}

float TinhTongDienTich(TamGiac *a, int n)
{
    // B1: tong = 0
    // B2: Duyệt từ đầu đến cuối mảng
    // B2.1: Cộng tổng với diện tích a[i]
}
```

- Yêu cầu 5: hàm đếm số tam giác có diện tích lớn hơn diện tích trung bình
  - o Gọi hàm tính tổng diện tích (câu 4)
  - o Gọi hàm tính diện tích cho từng a[i] của tam giác

```
int DemTamGiacLonHonDTichTBinh(TamGiac *a, int n)
{
    // B1: tb=TinhTongDienTich(a,n)/n;
    // B2: Gan dem: 0
    // B3: Duyệt từ đầu đến cuối mảng
    // B3.1: Nếu diện tích(a[i]) > tb
    // B3.1.1: dem ++
}
```

- Yêu cầu 6: hàm sắp xếp các tam giác giảm dần theo chu vi

```
void SapGiamTheoChuVi(TamGiac *a, int n)
```

```
{  
    // B1: i duyệt từ 0 đến n-2  
    // B1.1: j duyệt từ i+1 đến n-1  
    // B1.1.1: Nếu chu vi (a[i], 0) < chu vi (a[j], 0)  
    // B1.1.1.1: Hoán vị a[i] và a[j]  
}
```

### Bài 3

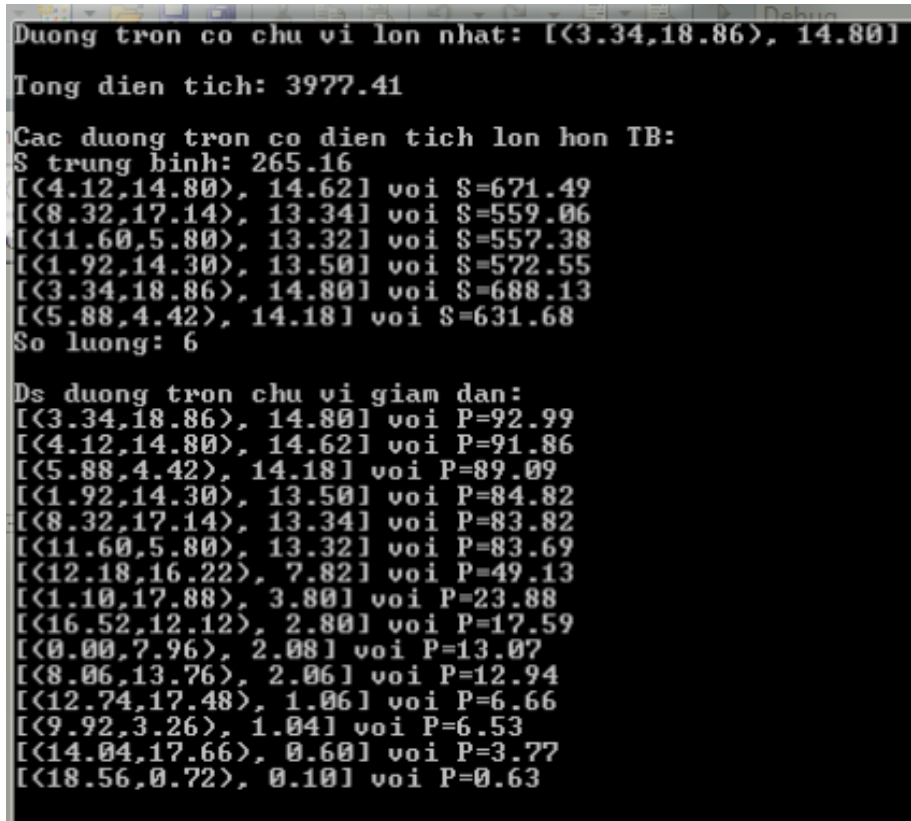
Xây dựng struct `DuongTron` như yêu cầu và hàm `main` sau chạy đúng:

```
void main()  
{  
    srand(4003);  
  
    // số lượng đường tròn  
    int n = 5 + rand()%20;  
  
    // tạo mảng các đường tròn  
    DuongTron* *listDT = (DuongTron**)malloc(n * sizeof(DuongTron*));  
    if (listDT == NULL)  
    {  
        return;  
    }  
    float xV, yV, rV;  
    for (int i=0; i<n; ++i)  
    {  
        xV = (rand()%1000)*1.0f/50;  
        yV = (rand()%1000)*1.0f/50;  
        rV = (rand()%1000+1)*1.0f/50;  
        listDT[i] = TaoDuongTron(xV, yV, rV);  
    }  
  
    // Xuất đường tròn có chu vi lớn nhất  
    printf("Duong tron co chu vi lon nhat: ");  
    DuongTron *pDTPMax = ChuViLonNhat(listDT, n);  
    PrintDuongTron(pDTPMax);  
    printf("\n\n");  
  
    // Tổng diện tích các đường tròn  
    printf("Tong dien tich: %0.2f", TongDienTich(listDT, n));  
    printf("\n\n");  
  
    // Xuất các đường tròn có diện tích lớn hơn diện tích trung bình  
    printf("Cac duong tron co dien tich lon hon TB:\n");  
    int dem = XuatCacDuongTronDTLonHonTB(listDT, n);  
    printf("So luong: %d", dem);  
    printf("\n\n");  
  
    // Xuất ds đường tròn giảm dần theo chu vi  
    printf("Ds duong tron chu vi giam dan:\n");  
    XuatDuongTronChuViGiam(listDT, n);  
    printf("\n\n");  
}
```

```
// thu hồi bộ nhớ
XoaDsDuongTron(listDT, n);

getch();
}
```

Kết quả:



```
Duong tron co chu vi lon nhat: [(3.34,18.86), 14.80]
Tong dien tich: 3977.41
Cac duong tron co dien tich lon hon TB:
S trung binh: 265.16
[(4.12,14.80), 14.62] voi S=671.49
[(8.32,17.14), 13.34] voi S=559.06
[(11.60,5.80), 13.32] voi S=557.38
[(1.92,14.30), 13.50] voi S=572.55
[(3.34,18.86), 14.80] voi S=688.13
[(5.88,4.42), 14.18] voi S=631.68
So luong: 6
Ds duong tron chu vi giam dan:
[(3.34,18.86), 14.80] voi P=92.99
[(4.12,14.80), 14.62] voi P=91.86
[(5.88,4.42), 14.18] voi P=89.09
[(1.92,14.30), 13.50] voi P=84.82
[(8.32,17.14), 13.34] voi P=83.82
[(11.60,5.80), 13.32] voi P=83.69
[(12.18,16.22), 7.82] voi P=49.13
[(1.10,17.88), 3.80] voi P=23.88
[(16.52,12.12), 2.80] voi P=17.59
[(0.00,7.96), 2.08] voi P=13.07
[(8.06,13.76), 2.06] voi P=12.94
[(12.74,17.48), 1.06] voi P=6.66
[(9.92,3.26), 1.04] voi P=6.53
[(14.04,17.66), 0.60] voi P=3.77
[(18.56,0.72), 0.10] voi P=0.63
```

Prototype:

```
#define PI 3.14156

struct Diem
{
    float x,y;
};

struct DuongTron
{
    Diem *tam;
    float banKinh;
};

DuongTron* TaoDuongTron(float xV, float yV, float rV);
void XoaDuongTron(DuongTron *p);
void PrintDuongTron(DuongTron *p);
DuongTron* ChuViLonNhat(DuongTron* *l, int n);
float TongDienTich(DuongTron* *l, int n);
```

## Thực hành Kỹ thuật lập trình

---

```
int XuatCacDuongTronDTLonHonTB(DuongTron* *l, int n);  
void XuatDuongTronChuViGiam(DuongTron* *l, int n);  
void XoaDsDuongTron(DuongTron* *l, int n);
```