

## السؤال الثاني: (10 عشرة علامات)

برنامج يقوم بتحديد القيمة الكبرى والقيمة الصغرى لقيمتين مدخلتين وذلك باستدعاء تابع مقارنة للقيم المدخلة يطلب استكمال وكتابة التابع اللازم ليتوافق مع مضمون جسم البرنامج الوارد أدناه .

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

int main () {
    int great,small,x,y;
    cout << "inter x=";
    cin>>x;
    cout<<"inter y=";
    cin>>y;
    compare(x,y,&great,&small);
    cout<<"great="<<great<<endl;
    cout<<"small="<<small<<endl;
    return 0;
}
```

## السؤال الثالث: (٣٠ ثلاثون علامة)

يتوفر ملف باسم mono.pxl يحوي معلومات صورة على شكل أغراض للنقط pixels . كل نقطة مكونة من محددات الاحداثيات x,y والألوان الخاصة بها R,G,B .  
يطلب كتابة برنامج يقوم بقراءة النقاط بالتسلسل من الملف من ثم تغيير درجات الألوان بمقدار ثابت يدخل من لوحة المفاتيح من ثم إعادة تسجيل النقاط في ملف جديد باسم stereo.pxl .  
عملية التعديل تتم من خلال عملية جمع معرفة كتحميل زائد على عملية الجمع مع قيمة ثابتة .  
النقطة تمثل على شكل صف يحوي الأعضاء:  
باني طبيعي ، تابع استرجاع المحددات ، تابع صديق للتحميل الزائد إضافة الى الهامد فقط .

## السؤال الأول: (30 ثلاثون علامة)

البرنامج التالي يحوي أخطاء برمجية فإذا علمت أن هذا البرنامج يستخدم لقراءة الأعداد العقدية من ملف على شكل غرض موصوف في البرنامج من ثم اراحة العدد بمقدار ثابت وتغيير قيمه أيضا بمقادير ثابتة ثم طباعة النواتج على شكل كتابة عقدية  $re + i im$  مع مطال الشعاع للحالات الجديدة والمطلوب كتابة البرنامج خالي من الأخطاء

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
class Complex {
private: double re; double im;
public:
    Complex(float r,float i){re=r,im(i)}
    Complex(float r) {re = r;im = 0.0;}
    ~Complex() {cout<<re;}
    double Magnitude(float *r,float &i);
    double Real() {return re;}
    double Imag();
    string toString();
    Complex operator+(Complex &b)
    {return Complex(re + *b.re,im + b.im);}
    Complex operator=(Complex b)
    {&re = b.re;im = *b.im; return this;}
};
double Complex::Imag(){ return *im;}
double Complex::Magnitude()
{ return sqrt(re*re + Imag()*Imag());}
String complex::toString()
{return strcat(re,strcpy('i',im));}
main() {
    ofstream fcmp;
    Fcmp.open("complexs.lex",ios::out);
    Complex *a;
    Complex *b = new Complex(1.0);
    Complex c,shc,newc;
    &a = new Complex(2.0,-2.0);
    c = a + b;
    while(fcmp.read((char*)&c,sizeof(*c))){
        shc=c+&b; newc=c+a; *a=&newc; c=shc;
        cout<<"complexA is ",a->toString(a),
        "MagnitudeA=",a.Magnitude()<<endl;
        cout<<"compleC is ",c.toString(&c),
        "Magnitude=",c->Magnitude(*c)<<endl;
    }
    delete a; delete &b;
}
```





```
while(fcmp.read((char*)&c,sizeof(c)){
    shc=c+*b; newc=c+*a; *a=newc; c=shc;
    cout<<"complexA is ",a->
    toString(),"Magnitude=",a->
    Magnitude()<<endl;
    cout<<"complexC is
    ",c.toString(),"Magnitude=",c.Magnitude()<
    <endl;
}
delete a;
delete b;
close(fcmp);}
```

### السؤال الثاني : (10 عشرة علامات)

برنامج يقوم بتحديد القيمة الكبرى والقيمة الصغرى لقيمتين مدخلتين وذلك باستدعاء تابع مقارنة للقيم المدخلة يطلب استكمال وكتابة التابع اللازم ليتوافق مع مضمون جسم البرنامج الوارد أدناه .

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

void compare(int a,int b,int
*pgreat,int *psmall)
{
    If (a>b) {*pgreat=a;*psmall=b;}
    Else {*pgreat=b;*psmall=a;}
}

int main () {
    int great,small,x,y;
    cout << "inter x=";
    cin>>x;
    cout<<"inter y=";
    cin>>y;
    compare(x,y,&great,&small);
    cout<<"great="<<great<<endl;
    cout<<"small="<<small<<endl;
    return 0;
}
```

### السؤال الأول : (30 ثلاثون علامة)

البرنامج التالي يحوي أخطاء برمجية فإذا علمت أن هذا البرنامج يستخدم لقراءة الأعداد العقدية من ملف على شكل  $re + i im$  مع مطال ثابتة ثم طباعة النواتج على شكل كتابة عقدية  $re + i im$  مع مطال الشعاع للحالات الجديدة والمطلوب كتابة البرنامج خالي من الأخطاء

```
#include <iostream>
#include <string>
#include <Math>
#include <fstream>
using namespace std;
class Complex {
private:
    double re; double im;
public:
    Complex(float r,float i):re(r),im(i){}
    Complex(float r) {re = r;im = 0.0;}
    ~Complex() {}
    double Magnitude();//calculate magnitude
    double Real() {return re;}
    double Imag();
    string toString();
    Complex operator+(Complex b)
    {return Complex(re + b.re,im + b.im);}
    Complex operator=(Complex b)
    {re = b.re;im = b.im; return *this;}
};
double Complex::Imag(){ return im;}
double Complex::Magnitude()
{ return sqrt(re*re + Imag()*Imag());}
String complex::toString()
{return strcat(re, strcat('+i', im));}
main()
{
    ifstream fcmp;
    fcmp.open("complexs.lex",ios::binary);
    Complex *a;
    Complex *b = new Complex(1.0);
    Complex c,shc,newc;
    a = new Complex(2.0,-2.0);
    c = *a + *b;
}
```

### السؤال الثالث : (30 ثلاثون علامة)

يتوفر ملف باسم mono.pxl يحوي معلومات صورة على شكل أغراض للنقط pixels . كل نقطة مكونة من محددات الاحداثيات x,y والألوان الخاصة بها R,G,B . يطلب كتابة برنامج يقوم بقراءة النقاط بالتسلسل من الملف من ثم تغيير درجات الألوان بمقدار ثابت يدخل من لوحة المفاتيح من ثم إعادة تسجيل النقاط في ملف جديد باسم stereo.pxl .



التعديل تتم من خلال عملية جمع معرفة كتحميل زائد على عملية الجمع مع قيمة ثابتة  
مخزنة تمثل على شكل صف يحوي الأعضاء: باني طبيعي وتابع استرجاع المحددات وتابع صديق  
تحميل الزائد إضافة الى الهادم فقط .

يوجد أكثر من حل لهذه المسألة منها تدور جميعها حول الحل المعروض ادناه:

```
# include<iostream>
#include<fstream>
Using namespace std;
Class pixel {
Private:
Int x,y,r,g,b;
Public:
Pixel(int xi,int yi,int ri,int gi,int bi) {
x=xi,y=yi,r=ri,g=gi,b=bi;
}
Void getpixel(int *xo,int *yo,int *ro,int *go,int *bo) {
*xo=x; *yo=y;*ro=r;*go=g;*bo=b;
}
~ pixel();
Friend pixel operator + (pixel ob,int i)
{
Pixel temp;
Temp.r=ob.r+i; temp.g=ob.g+i; temp.b=ob.b+i;temp.x=ob.x;temp.y=ob.y;
Return temp;
}
Int main() {
Ofstream ofpixel("stereo.pxl",ios::binary);
Ifstream ifpixel("mono.pxl",ios::binary);
Pixel pout,pin;
Int delta;
Cout<<"insert update value=";
Cin>>delta;
While (ifpixel.read((char *)& pin,sizeof(pin)))
{
Pout=pin+delta;
Ofpixel.write((char *)&pout,sizeof(pout));
}
Close(ofpixel); close(ifpixel);
}
```

مدرس المادة د مفيد حداد

2019/01/14