

Examen du 14/10/2002

Corrigé

Exercice I

1.

```
typedef struct compl {
    float re, im;
} complexe;

complexe CreerComplexe (float a, float b)
    /* Creation d'un nombre complexe
ENTREE : a partie reelle et b partie imaginaire de type float
SORTIE : a+ib de type complexe
    */
{
    complexe z;
    z.re=a;
    z.im=b;
    return z;
}

float PartieReelle (complexe z)
    /* Partie reelle d'un nombre complexe
ENTREE : z de type complexe
SORTIE : partie reelle de z, de type float
    */
{
    return(z.re);
}

float PartieImaginaire (complexe z)
    /* Partie imaginaire d'un nombre complexe
ENTREE : z de type complexe
SORTIE : partie imaginaire de z, de type float
    */
{
    return(z.im);
}

complexe SommeComplexe (complexe z1, complexe z2)
    /* Somme de deux nombres complexes
ENTREE : z1 et z2 de type complexe
SORTIE : z1+z2 de type complexe
    */
{
    float a, b;
```

```

    complexe z;
    a=PartieReelle(z1)+PartieReelle(z2);
    b=PartieImaginaire(z1)+PartieImaginaire(z2);
    z=CreerComplexe(a,b);
    return z;
}

complexe ProduitComplexe (complexe z1, complexe z2)
    /* Produit de deux nombres complexes
ENTREE : z1 et z2 de type complexe
SORTIE : z1*z2 de type complexe
    */
{
    float a, b;
    complexe z;
    a=(PartieReelle(z1))*(PartieReelle(z2))-(PartieImaginaire(z1))*(PartieImaginaire(z2));
    b=(PartieReelle(z1))*(PartieImaginaire(z2))+(PartieImaginaire(z1))*(PartieReelle(z2));
    z=CreerComplexe(a,b);
    return z;
}

float ModuleComplexe (complexe z)
    /* Module d'un nombre complexe
ENTREE : z de type complexe
SORTIE : |z| de type float
    */
{
    return(sqrt((PartieReelle(z))*(PartieReelle(z))+(PartieImaginaire(z))*(PartieImaginaire(z))));
}

void AfficherComplexe (complexe z)
    /* Affichage d'un nombre complexe
ENTREE : z de type complexe
EFFET : affichage du complexe sur stdout
    */
{
    if (PartieImaginaire(z)>0)
        printf("%.5f+%.5fi\n",PartieReelle(z),PartieImaginaire(z));
    if (PartieImaginaire(z)<0)
        printf("%.5f-%.5fi\n",PartieReelle(z),fabs(PartieImaginaire(z)));
    if (PartieImaginaire(z)==0)
        printf("%.5f\n",PartieReelle(z));
}

2.

int EstDansM (complexe K)
    /* Fonction indicatrice de l'ensemble de Mandelbrot
ENTREE : un complexe K
SORTIE : 1 si K est dans l'ensemble de Mandelbrot, 0 sinon.
REMARQUE : On suppose qu'une suite de complexes est bornee ssi
les modules des 250 premiers termes sont inferieurs a 4
    */
{

```

```

complexe z; // contient le zn
float module, MaxModule; // module et plus grand module
int n; // variable de boucle

z=CreerComplexe(0.0,0.0);
MaxModule=0.0;
for (n=1;n<=250;n++)
{
    z=SommeComplexe(ProduitComplexe(z,z),K);
    module=ModuleComplexe(z);
    if (module>MaxModule) MaxModule=module;
}
return (MaxModule<=4);
}

```

3.

```

int main()
{
    float x1, x2, y1, y2;
    int n1, n2, j, l;
    complexe z;

    printf("Entrez x1 x2 y1 et y2 separees par des espaces\n");
    scanf("%f %f %f %f",&x1,&x2,&y1,&y2);
    printf("Entrez n1 et n2 separees par des espaces\n");
    scanf("%d %d",&n1,&n2);

    for (l=0;l<=n2;l++)
        for (j=0;j<=n1;j++)
    {
        z=CreerComplexe(x1+j*(x2-x1)/n1,y1+l*(y2-y1)/n2);
        if (EstDansM(z)) AfficherComplexe(z);
    }

    return 0;
}

```

Exercice II

Le programme affiche 2 2 car z est un tableau dont chaque case point vers x . La modification de x entraine donc la modification des deux cases du tableau.