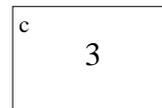
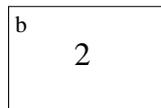
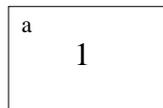


Examen du 13/11/2007

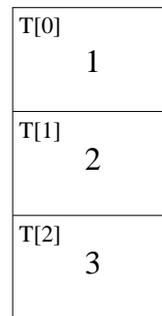
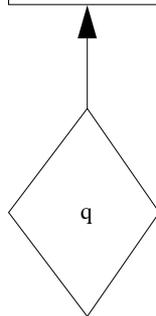
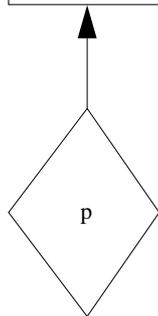
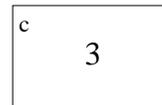
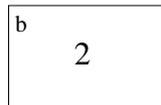
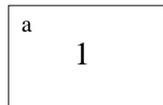
Corrigé

Exercice I

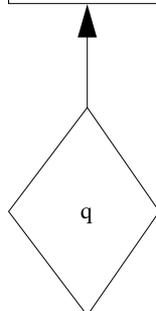
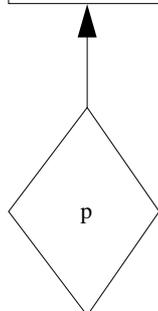
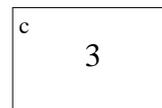
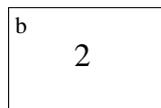
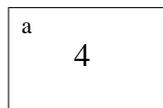
A l'initialisation des variables, on a



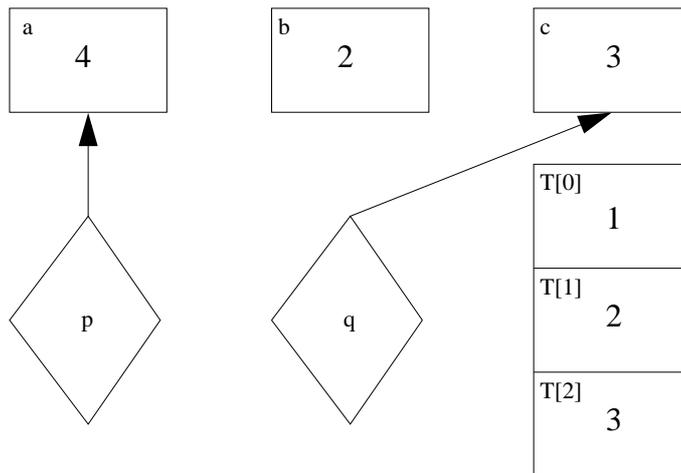
Après les deux premières lignes de code on a



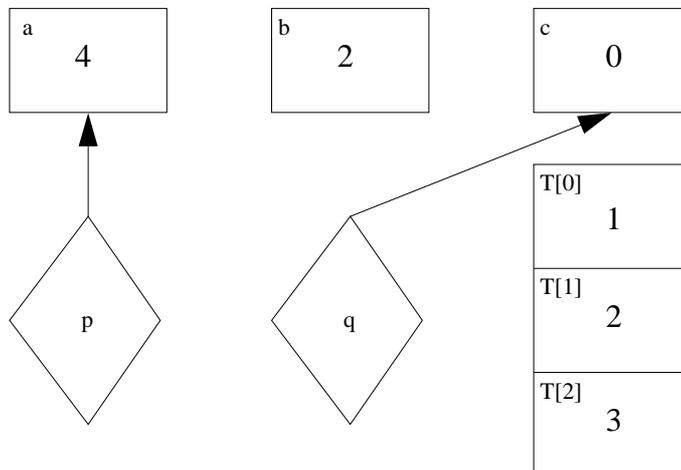
L'affectation suivante conduit à



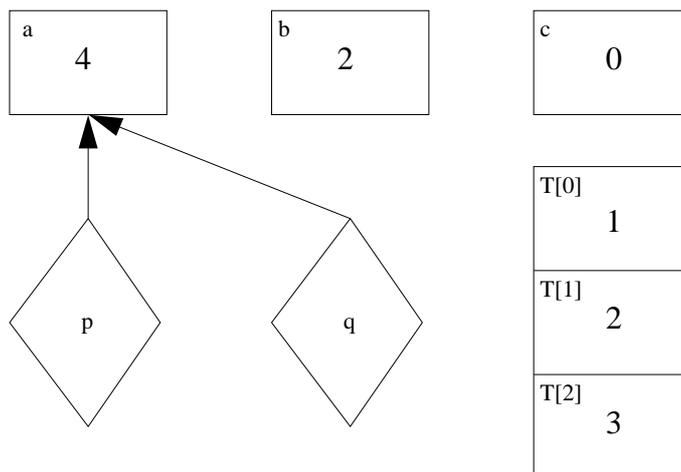
On obtient alors l'affichage [1] 4 2 3 4 1. L'affectation suivante donne



On obtient alors l'affichage [2] 4 2 3 3 2. L'affectation suivante donne



On obtient alors l'affichage [3] 4 2 0 0 2.



On obtient alors l'affichage [4] 4 2 0 4 3.

Exercice II

Le programme suivant répond à l'exercice

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

int main() {
    int Bareme, NombreDeNotes, i, Note, *Frequence;

    /* Demande du bareme et du nombre de notes */

    do {
        printf("Entrer le nombre de points sur lequel est notee l'epreuve\n");
        scanf("%d",&Bareme);
    } while (Bareme<=0);

    do {
        printf("Entrer le nombre de notes\n");
        scanf("%d",&NombreDeNotes);
    } while (NombreDeNotes<=0);

    /* Creation et initialisation du tableau destine a compter la frequence
       des notes */

    Frequence=(int*)malloc((Bareme+1)*sizeof(int));

    for (i=0;i<=Bareme;i++)
        Frequence[i]=0;

    /* Demande des notes et mise a jour du tableau des frequences */

    for (i=0;i<NombreDeNotes;i++) {

        do {
            printf("Entrer la note no %d\n",i);
            scanf("%d",&Note);
        } while ((Note<0)|| (Note>Bareme));

        Frequence[Note]++;

    }

    /* Affichage de la table */

    for (i=0;i<=Bareme;i++)
        printf("La note %d a ete entree %d fois\n",i,Frequence[i]);

    return 0;
}
```

Exercice III

1. Le code suivant répond à la question

```
typedef struct H_struct { double a, b, c, d; } H;
```

2. Le code suivant répond à la question

```
H CreerH (double a, double b, double c, double d)
/* Creation d'une fonction homographique
   ENTREE : les coefficients a, b, c, d
   SORTIE : la fonction homographique correspondante
*/
{
    H f;

    f.a = a;
    f.b = b;
    f.c = c;
    f.d = d;

    return f;
}
```

3. Le code suivant répond à la question

```
int EstAffineH (H f)
/* Predicat indiquant si f est affine
   ENTREE : la fonction homographique f
   SORTIE : 1 si f est affine, 0 sinon
*/
{
    return((f.b==0)&&(f.d==0));
}
```

4. Le code suivant répond à la question

```
double PointSingularH (H f)
/* Determination du point singulier d'une fonction homographique non constante
   ENTREE : une fonction homographique f non constante
   SORTIE : le reel ou elle n'est pas defini
*/
{
    assert(f.c!=0);
    return(-f.d/f.c);
}
```

5. Le code suivant répond à la question

```
double CalculerH (H f, double x)
/* Evaluation d'une fonction homographique en un point
   ENTREE : la fonction f et le reel x
   SORTIE : f(x)
*/
```

```

{
double numerateur = f.a*x+f.b;
double denominateur = f.c*x+f.d;

assert(denominateur!=0);
return(numerateur/denominateur);
}

```

6. Soit x un point non singulier de f et notons $y = f(x)$ alors les ligne ssuivantes sont équivalentes

$$\begin{aligned}
y &= \frac{ax + b}{cx + d} \\
cxy + dy &= ax + b \\
cxy - ax &= -dy + b \\
x(cy - a) &= -dy + b \\
x &= \frac{-dy + b}{cy - a}
\end{aligned}$$

Cela consuit au code suivant

```

H InverseH (H f)
/* Calcul de la reciproque d'une fonction homographique
   ENTREE : la fonction homographique f
   SORTIE : son inverse
*/
{
  H finv;

  finv.a = -f.d;
  finv.b = f.b;
  finv.c = f.c;
  finv.d = -f.a;

  return finv;
}

```

7. Soit f_1 et f_2 deux fonctions homographiques, données par

$$f_1(x) = \frac{a_1x + b_1}{c_1x + d_1} \quad \text{et} \quad f_2(x) = \frac{a_2x + b_2}{c_2x + d_2}$$

on a

$$\begin{aligned}
f_1 \circ f_2(x) &= f_1(f_2(x)) \\
&= \frac{a_1 f_2(x) + b_1}{c_1 f_2(x) + d_1} \\
&= \frac{a_1 \frac{a_2x + b_2}{c_2x + d_2} + b_1}{c_1 \frac{a_2x + b_2}{c_2x + d_2} + d_1} \\
&= \frac{\frac{a_1 a_2 x + a_1 b_2 + b_1 c_2 x + b_1 d_2}{c_2 x + d_2}}{\frac{c_1 a_2 x + c_1 b_2 + d_1 c_2 x + d_1 d_2}{c_2 x + d_2}} \\
&= \frac{(a_1 a_2 + b_1 c_2)x + (a_1 b_2 + b_1 d_2)}{(c_1 a_2 + d_1 c_2)x + (c_1 b_2 + d_1 d_2)}
\end{aligned}$$

Cela consuit au code suivant

```

H ComposeH (H f1, H f2)
/* Calcul de la composee de deux fonctions homographiques
   ENTREE : f1 et f2
   SORTIE : f1 o f2
*/
{
  H fcomp;
  double a1 = f1.a;
  double b1 = f1.b;
  double c1 = f1.c;
  double d1 = f1.d;
  double a2 = f2.a;
  double b2 = f2.b;
  double c2 = f2.c;
  double d2 = f2.d;

  fcomp.a = a1*a2+b1*c2;
  fcomp.b = a1*b2+b1*d2;
  fcomp.c = c1*a2+d1*c2;
  fcomp.d = c1*b2+d1*d2;

  return fcomp;
}

```

8. Soit f une fonction homographique définie par

$$f(x) = \frac{ax + b}{cx + d}$$

Cette fonction est le quotient de deux fonctions dérivables, elle est dérivable sur l'ensemble des points où le dénominateur ne s'annule pas. On a

$$f'(x) = \frac{a(cx + d) - (ax + b)c}{(cx + d)^2}$$

ce qui donne

$$f'(x) = \frac{ad - bc}{(cx + d)^2}$$

Dès que c est non nul, cette fonction n'est pas homographique.

9. Soit f_1 et f_2 deux fonctions homographiques, données par

$$f_1(x) = \frac{a_1x + b_1}{c_1x + d_1} \quad \text{et} \quad f_2(x) = \frac{a_2x + b_2}{c_2x + d_2}$$

on a $f_1 = f_2$ si et seulement si pour tout réel x on a

$$\frac{a_1x + b_1}{c_1x + d_1} = \frac{a_2x + b_2}{c_2x + d_2}$$

ce qui équivaut à : $x \notin \{-\frac{d_1}{c_1}, -\frac{d_2}{c_2}\}$ (si c_1 ou c_2 est nul on ignore le réel non défini) et

$$(a_1x + b_1)(c_2x + d_2) = (a_2x + b_2)(c_1x + d_1)$$

c'est-à-dire

$$a_1c_2x^2 + (b_1c_2 + a_1d_2)x + b_1d_2 = a_2c_1x^2 + (a_2d_1 + b_2c_1)x + d_1b_2$$

il faut et suffit alors que les coefficients de ces deux polynômes soient égaux, c'est-à-dire

$$\begin{cases} a_1c_2 = a_2c_1 \\ b_1c_2 + a_1d_2 = a_2d_1 + b_2c_1 \\ b_1d_2 = d_1b_2 \end{cases}$$

Ce qui conduit au code suivant

```
int EstEgalH (H f1, H f2)
/* Predicat d'egalite de deux fonctions homographiques
   ENTREE : les deux fonctions f1 et f2
   SORTIE : 1 si les fonctions sont egales, 0 sinon
*/
{
    double a1 = f1.a;
    double b1 = f1.b;
    double c1 = f1.c;
    double d1 = f1.d;
    double a2 = f2.a;
    double b2 = f2.b;
    double c2 = f2.c;
    double d2 = f2.d;

    return ((a1*c2==a2*c1)&&(b1*c2+a1*d2==a2*d1+c1*b2)&&(b1*d2==b2*d1));
}
```

10. On propose le programme suivant

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

ON MET ICI LE CODE DE LA QUESTION 1

ON MET ICI LE CODE DES QUESTIONS 2, 3, 4, 5, 6, 7 ET 9

int main()
{
    H f1, f2, g;
    double a=1, b=2, c=3, d=4;
    double x=2.1;

    f1 = CreerH(a,b,c,d);
    f2 = CreerH(a,0,c,0);

    if (EstAffineH(f1)) printf("f1 est affine Affine\n");
    if (EstAffineH(f2)) printf("f2 est affine Affine\n");

    if ((f1.b!=0)&&(f1.d!=0))
        printf("f1 n'est pas definie en %f\n",PointSingulierH(f1));

    printf("f1(%f)=%f\n",x,CalculerH(f1,x));

    f2 = InverseH(f1);
    g = ComposeH(f1,f2);
```

```
if (EstEgalH(g,CreerH(1,0,0,1))) printf("Tout va bien\n");
else printf("Il y a un probleme !\n");

return(0);
}
```