Université Paris-Nord — L1
Institut Galilée — Année 2008-2009
Licence 1 - 2ème semestre

TD/TP1 : Structures et fonctions : les rationnels.

Nous travaillons ici sur les nombres rationnels. Un nombre rationnel sera représenté par une structure `rat` à deux champs : un champ `num` représentant le numérateur et un champ `den` représentant le dénominateur d'une fraction. Nous noterons par la suite dans les exemples (a/b) un rationnel représenté sous cette forme.

1 Structures et rationnels

1. Définir le type `rat`.
2. Écrire les fonctions suivantes :
   (a) `lit_rat` lit au clavier le numérateur et le dénominateur d'une fraction et renvoie le rationnel (un objet de type `struct rat`) correspondant.
   (b) `somme_rat` renvoie la somme de deux rationnels `r1` et `r2`.
   (c) `produit_rat` renvoie le produit de deux rationnels `r1` et `r2`.
   (d) `affiche_rat` affiche le numérateur et le dénominateur d'un rationnel `r`.
3. Ecrire un programme qui lit trois rationnels `r1`, `r2`, `r3`, calcule la somme de `r1` et de `r2` et calcule et affiche le produit de `r1+r2` par `r3`.
4. Ajouter les fonctions `difference_rat` et `quotient_rat` et modifier le programme pour qu'il calcule également la différence de `r1` et de `r2` et affiche le quotient de `r1-r2` par `r3`.
5. Simuler le programme pour le calcul de (5/2 + 5/2) * 4/2 avant et après cette modification.

Correction. Correction Note aux chargés de TD : ils n'ont pas vu typedef donc : NE PAS UTILISER

```
  typedef struct RAT_T
{
        int num;
        int den;
} rat_t;
```

donc remplacer PARTOUT DANS LE CODE `rat_t` PAR `struct RAT_T`

```
# include <stdio.h>

/* type */
typedef struct RAT_T
{
```

```
        int num;
        int den;
} rat_t;

/* prototypes */
rat_t lit_rat(void);
void affiche_rat(rat_t c);
rat_t somme_rat(rat_t c1, rat_t c2);
rat_t produit_rat(rat_t c1, rat_t c2);

/* principale */
int main()
{
        rat_t r1,r2,r3,s,p,res;
        r1=lit_rat();
        r2=lit_rat();
        r3=lit_rat();
        s=somme_rat(r1,r2);
        p=produit_rat(s,r3); /* ou p=produit (somme(r1,r2),r3) */
        affiche_rat(p);
        return 0;
}
/*fonctions */
rat_t lit_rat(void)
{
        rat_t c;
        printf(" (deux entiers a et b pour le rationnel a/b)\n");
        scanf("%d",&c.num);
        scanf("%d",&c.den);
        return (c);
}
void affiche_rat(rat_t c)
{
        printf("(%d/%d)\n",c.num,c.den);
        return ;
}
rat_t somme_rat(rat_t c1, rat_t c2)
{
        rat_t c;
        c.num=c1.num*c2.den + c2.num*c1.den ;
        c.den=c1.den*c2.den;
        return (c);
}
rat_t produit_rat(rat_t c1, rat_t c2)
{
        rat_t c;
        c.num=c1.num * c2.num;
        c.den=c1.den * c2.den;
        return (c);
}
```

```
//[Session started at 2007-11-07 18:11:11 +0100.]
//(deux entiers a et b pour le rationnel a/b)
//5 2
//(deux entiers a et b pour le rationnel a/b)
//5 2
//(deux entiers a et b pour le rationnel a/b)
//4 2
//(80/8)

/* declaration de fonctionnalites supplementaires */

#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf, scanf */

/* declaration constantes et types utilisateurs */



/* type */
typedef struct RAT_T
{
        int num;
        int den;
} rat_t;

/* declaration de fonctions utilisateurs */
int pgcd (int a, int b);
rat_t lit_rat(void);
void affiche_rat(rat_t c);
rat_t somme_rat(rat_t c1, rat_t c2);
rat_t difference_rat(rat_t c1, rat_t c2);
rat_t produit_rat(rat_t c1, rat_t c2);
rat_t quotient_rat(rat_t c1, rat_t c2);
rat_t rend_irreductible (rat_t r);

/* principale */
int main()
{
        rat_t r1,r2,r3,s,p,res;
        r1=lit_rat();
        r2=lit_rat();
        r3=lit_rat();
        s=somme_rat(r1,r2);
        p=produit_rat(s,r3); /* ou p=produit (somme(r1,r2),r3) */
        affiche_rat(p);
        return 0;
}
/*fonctions */
int pgcd (int a, int b){
        int aux,r;
        if (a<b){
```

```
                aux=a;
                a=b;
                b=aux;
        }
        r=a % b;
        while (r!=0){
                a=b;
                b=r;
                r=a%b;
        }
        return b;
}
rat_t lit_rat(void){
        rat_t c;
        printf(" (deux entiers a et b pour le rationnel a/b)\n");
        scanf("%d",&c.num);
        scanf("%d",&c.den);
        return (c);
}
void affiche_rat(rat_t c){
        printf("(%d/%d)\n",c.num,c.den);
        return ;
}
rat_t somme_rat(rat_t c1, rat_t c2){
        rat_t c;
        c.num=c1.num*c2.den + c2.num*c1.den ;
        c.den=c1.den*c2.den;
        c= rend_irreductible(c);
        return (c);
}
rat_t difference_rat(rat_t c1, rat_t c2){
        rat_t c;
        c.num=c1.num*c2.den - c2.num*c1.den ;
        c.den=c1.den*c2.den;
        c= rend_irreductible(c);
        return (c);
}
rat_t produit_rat(rat_t c1, rat_t c2)
{
        rat_t c;
        c.num=c1.num * c2.num;
        c.den=c1.den * c2.den;
        c= rend_irreductible(c);
        return (c);
}
rat_t quotient_rat(rat_t c1, rat_t c2){
        rat_t c;
        c.num=c1.num * c2.den;
        c.den=c1.den * c2.num;
        c= rend_irreductible(c);
        return (c);
```

```
}
rat_t rend_irreductible( rat_t r)
{
        /* utilise pgcd */
        int diviseur;
        rat_t res;
        if (r.num==0)
        {
                res.num=0;res.den=1;
                return res;
        }
        diviseur=pgcd( r.num, r.den);
        res.num= r.num / diviseur;
        res.den= r.den / diviseur;
        return res;
}


//[Session started at 2007-11-07 18:30:54 +0100.]
//(deux entiers a et b pour le rationnel a/b)
//5 2
//(deux entiers a et b pour le rationnel a/b)
//5 2
//(deux entiers a et b pour le rationnel a/b)
//4 2
//(10/1)
```

2 TP : Mini-calculatrice

Il s'agit ici d'écrire un programme qui demande à l'utilisateur d'entrer une expression simple à évaluer et qui affiche la valeur de l'expression [1]. L'expression à saisir concerne des nombres rationnels et suit la forme suivante : `nombre rationnel opérateur nombre rationnel`, avec `opérateur` étant un des quatre opérateurs arithmétiques '+', '-', '*' ou '/'. Deux exemples d'exécution sont les suivants :

```
Entrez une expression de la forme : nombre operateur nombre
15 9 * 2 1
15/9 * 2/1  = 30/9
```

1. Implanter le programme de calcul sur les rationnels du td et le tester sur plusieurs exemples.
2. Ajouter la fonction `pgcd` (appel par `pgcd (a,b)`) en prenant le principe suivant :

   pgcd des entiers positifs a et b :

   si a < b alors echanger a et b

   initialiser une variable r a a modulo b
   tant que r est different de 0

1. Il s'agit de la calculette vue au TP6 du semestre 1

```
                mettre la valeur de b dans a
                mettre la valeur de r dans b
                mettre la valeur de a modulo b dans r ;
            fin tant que
            renvoyer b
    }
```

On essaiera l'algorithme sur des exemples simples pour pouvoir tester ensuite la fonction `pgcd` implantée.

3. Ajouter au programme une fonction `rend_irreductible` telle que `rend_irreductible(r)` renvoie un rationnel équivalent mais sous la forme d'une fraction irréductible. Ainsi soit le rationnel `r` = (4/2), alors `rend_irreductible(r)` renvoie le rationnel (2/1). On utilisera pour cela la fonction `pgcd` dans le corps de `rend_irreductible`.
4. Modifier les fonctions `difference_rat` et `quotient_rat`, `lit_rat`, `somme_rat` et `produit_rat` de manière à ce que le résultat soit toujours sous forme irréductible.

Correction. Correction

```
/* declaration de fonctionnalites supplementaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf, scanf */

/* declaration constantes et types utilisateurs */
typedef struct RAT_T
{
        int num;
        int den;
} rat_t;

/* declaration de fonctions utilisateurs */
int pgcd (int a, int b);
rat_t lit_rat(void);
void affiche_rat(rat_t c);
rat_t somme_rat(rat_t c1, rat_t c2);
rat_t difference_rat(rat_t c1, rat_t c2);
rat_t produit_rat(rat_t c1, rat_t c2);
rat_t quotient_rat(rat_t c1, rat_t c2);
rat_t rend_irreductible (rat_t r);

 ......

/* fonction principale */
int main()
{
    rat_t nombre_g; /* membre gauche de l'expression */
    rat_t  nombre_d; /* membre droit de l'expression */
    char op; /* operateur */
    rat_t expr; /* resultat de l'expression */
```

```
    /* saisie expression */
    printf("les nombres ici sont rationnels. Pour saisir un nombre saisir le numerateur puis le denominate
    printf("Entrez une expression de la forme : nombre operateur nombre\n");
    nombre_g=lit_rat();
    scanf(" %c",&op);
    nombre_d=lit_rat();


    /* calcul valeur expression */
    /* cas mutuellement exclusif */
    if(op == '+') /* addition */
    {
        expr = somme_rat (nombre_g , nombre_d);
    }

    if(op == '-') /* soustraction */
    {
        expr = difference_rat (nombre_g , nombre_d);
    }

    if(op == '*') /* multiplication */
    {
       expr =produit_rat (nombre_g , nombre_d);

    }

    if(op == '/') /* division */
    {
        expr =quotient_rat (nombre_g , nombre_d);
    }

    /* affichage resultat */
    affiche_rat(expr);

    return EXIT_SUCCESS;
}

/* implantation de fonctions utilisateurs */
```