POO
Correction du TD2

Exercice 1

```
//prime_familiale.h

class Prime_Familiale
{
	float p_situation_familiale,
		p_enfant,
		p_habitation_sociale,
		p_transport;

public:
	Prime_Familiale();
	virtual ~Prime_Familiale();
	void Changer_prim_sit_fam (float);
	void Changer_prime_enfants (float);
	void Changer_prime_habt (float);
	void Changer_prim_transp (float);
	float Calcul_prime ();
};

////////////////////////////////////////////////////////
//prime_familiale.cpp

#include "Prime_Familiale.h"
Prime_Familiale::Prime_Familiale()
{

	p_situation_familiale=0;
	p_enfant=0;
	p_habitation_sociale=0;
	p_transport=0;
}

Prime_Familiale::~Prime_Familiale()
{

}

void Prime_Familiale::Changer_prim_sit_fam (float p)
{
	p_situation_familiale=p;
}

void Prime_Familiale::Changer_prime_enfants (float p)
{
	p_enfant=p;
```

```
}
void Prime_Familiale::Changer_prime_habt (float p)
{
	p_habitation_sociale=p;
}

void Prime_Familiale::Changer_prim_transp (float p)
{
	p_transport=p;
}
float Prime_Familiale::Calcul_prime ()
{
	float p = p_situation_familiale+ p_enfant+ p_habitation_sociale+ p_transport;
	if (p<0)
		p=0;
	return p;
}
```

Exercice 2

```
//Set_let.h
class set_let
{

	char t[52];
	int card;
public:
	set_let() //C'est une méthode en ligne (inline)
	{
		card=0;

	}
	virtual ~set_let();

	void operator << (char c); //Ajouter un élément à la fin du tableau
	void operator >> (char c);//enlever le caractère c  du tableau
	int operator ! ();

};

//////////////////////////////////////////////////////
//set_let.cpp
#include "set_let.h"
#include <iostream.h>

set_let::~set_let()
{

}
```

```
void set_let::operator << (char c)
{
	if (card <52) //Il ne faut pas dépasser la taille maximale
	{
		card++;
		t[card] = c;
	}
	else
		cout <<"tableau plein, impossible d'ajouter";
}
void set_let::operator >> (char c)
{
	int pos=0;
	//Il faut d'abord chercher la position du caractèe c

	while ((t[pos]!=c) && (pos<52))
	{
		pos++;

	}

	if (pos==52) //Element non trouvé

		cout <<"Element introuvable";
	else
	{
		t[pos] = '\0'; // On met le carctère vide à la place du caractère enlevé
		card--;		// On décrémente e cardinal
	}

}
int set_let::operator ! ()
{
	return card;
}

//////////////////////////////////////////////////////
//Exemple d'appel dans le programme principal

void main()
{
	set_let s;
	cout <<!s;	//Affiche le cadinale de s, le résultat est 0
	s <<'Z';	//Ajouter la lettre Z;
	s <<'p';//Ajouter la lettre p;
	s >> 'Z';	//Enlever la lettre Z
	cout <<!s;	//Affiche le cadinale de s, le résultat est 1
}
```

Exercice 4

```
//rationel.h

class CRationel
{
	int numerateur, denominateur;

public:

	CRationel(int n=0, int d=1); //Constructeur surchargé

	void Creer_Rationel();

	void Afficher_rationel();
	void Afficher_reel();

	CRationel operator +(CRationel r);
	CRationel operator -(CRationel r);
	CRationel operator *(CRationel r);
	CRationel operator /(CRationel r);
	bool operator <(CRationel r);
	bool operator <=(CRationel r);
	bool operator >(CRationel r);
	bool operator >=(CRationel r);
	bool operator ==(CRationel r);
	bool operator !=(CRationel r);

	virtual ~CRationel();

};

////////////////////////////////////
//rationel.cpp
#include "Rationel.h"
#include <iostream.h>


CRationel::~CRationel()
{

}

CRationel::CRationel(int n, int d)
{
	numerateur=n;
	denominateur=d;
	Creer_Rationel();//Mettre le rationel sous sa forme réduite
}

void CRationel::Creer_Rationel()
```

```
{
	//Calculer le pgcd du numérateur et du dénominateur pour avoir la forme réduite
	int t;

	if (numerateur < denominateur)
		t = numerateur;
	else
		t = denominateur;

	while ((numerateur % t) != 0 || (denominateur % t) != 0)
		t = t-1;

	numerateur=numerateur/t;
	denominateur=denominateur/t;

}

void CRationel::Afficher_rationel()
{
	cout <<numerateur<<"/"<<denominateur;
}
void CRationel::Afficher_reel()
{
	float r = (float)numerateur/denominateur;
	cout <<r;
}


CRationel CRationel::operator +(CRationel r)
{
	int num = numerateur*r.denominateur + r.numerateur*numerateur;
	int den = denominateur*r.denominateur;
	CRationel res (num, den); //Appel au constructeur ,
				  //qui met le rationel res sous sa forme réduite
	return res;
}
CRationel CRationel::operator -(CRationel r)
{
	int num = numerateur*r.denominateur - r.numerateur*numerateur;
	int den = denominateur*r.denominateur;
	CRationel res (num, den); //Appel au constructeur ,
				   //qui met le rationel res sous sa forme réduite
	return res;
}

CRationel CRationel::operator *(CRationel r)
{
	int num = numerateur * r.numerateur;
	int den = denominateur*r.denominateur;
```

```
        CRationel res (num, den); //Appel au constructeur ,
                                          //qui met le rationel res sous sa forme réduite
        return res;
}

CRationel CRationel::operator /(CRationel r)
{
        int num = numerateur * r.denominateur;
        int den = denominateur*r.numerateur;

        CRationel res (num, den); //Appel au constructeur ,
                                        //qui met le rationel res sous sa forme réduite
        return res;

}

bool CRationel::operator <(CRationel r)
{
        int num1= numerateur*r.denominateur;
        int num2= r.numerateur*numerateur;

        if (num1<num2)
                return true;
        else
                return false;

}

bool CRationel::operator <=(CRationel r)
{
        int num1= numerateur*r.denominateur;
        int num2= r.numerateur*numerateur;

        if (num1<=num2)
                return true;
        else
                return false;
}

bool CRationel::operator >(CRationel r)
{
        int num1= numerateur*r.denominateur;
        int num2= r.numerateur*numerateur;

        if (num1>num2)
                return true;
        else
                return false;
}
```

```
bool CRationel::operator >=(CRationel r)
{
	int num1= numerateur*r.denominateur;
	int num2= r.numerateur*numerateur;

	if (num1>=num2)
		return true;
	else
		return false;
}

bool CRationel::operator ==(CRationel r)
{
	int num1= numerateur*r.denominateur;
	int num2= r.numerateur*numerateur;

	if (num1==num2)
		return true;
	else
		return false;
}

bool CRationel::operator !=(CRationel r)
{
	int num1= numerateur*r.denominateur;
	int num2= r.numerateur*numerateur;

	if (num1==num2)
		return true;
	else
		return false;
}
////////////////////////////////////////
//Exemple d'appel

void main()
{
	CRationel z(4,6);
	z.Afficher_rationel(); //le résultat est 2/3
	CRationel r(4,5);
	r.Afficher_rationel(); //Le résultat est 4/5
	r.Afficher_reel();		//Le résultat est 0.8
	CRationel s(1,2);

	CRationel t = r+s;

	t.Afficher_rationel();//Le résultat est 1/5
	if (s==t)
		cout <<"ils sont égaux";
}
```

Exercice 5

```
/ /livre.h

class Livre
{

public:

        char* titre;
        char* auteur;
        int nb_exp;
        int annee_edition;

        Livre();
        Livre(char* t, char* aut, int n=0, int an=2001);
        void Changer_auteur(char* nom);
        void Changer_titre(char* nvtitre);
        void Augmenter_exemplaires (int nb);
        void Diminuer_exemplaires (int nb);
        void Affichr_livre();
        virtual ~Livre();

};

////////////////////////////////////////// :
//Livre.cpp

#include "Livre.h"
#include <iostream.h>
Livre::Livre()
{
        titre="";
        auteur="";
        nb_exp=0;
        annee_edition=0;

}

Livre::~Livre()
{

}

Livre::Livre(char* t, char* aut, int n, int an)
{
        auteur=aut;
        titre=t;
        nb_exp=n;
```

```
}
void Livre::Changer_auteur(char* nom)
{
	auteur=nom;
}
void Livre::Changer_titre(char* nvtitre)
{
	titre=nvtitre;
}
void Livre::Augmenter_exemplaires (int nb)
{
	nb_exp+=nb;

}
void Livre::Diminuer_exemplaires (int nb)
{
	nb_exp-=nb;
}
void Livre::Affichr_livre()
{
	cout <<"titre:"<<titre;
	cout <<"auteur:"<<auteur;
	cout <<"nombre d'exemplaires:"<<nb_exp;
	cout <<"année d'édition:"<<annee_edition;
}
```

Exercice 6

```
//personne.h
#include <iostream.h>
class Personne
{
	char* nom, *prenom;
	int age;

public:  //Toutes les méthodes sont en ligne
	Personne()
	{
		nom = new char[255];
		prenom = new char [255];
	}

	Personne (char* n, char* p)
	{
		nom = new char[sizeof(n)];
		prenom = new char [sizeof(p)];
		nom= n;
		prenom = p;
	}
```

```
        void Changer_nom(char* nvnom,char* nvprenom)
        {
                nom=nvnom;
                prenom = nvprenom;

        }
        void Changer_age (int nb)
        {
                age = nb;
        }

        void Dire_bonjour ()
        {
                cout <<"Je suis" << nom << prenom<<"j'ai"<<age <<"ans";
        }

        virtual ~Personne()
        {
                delete nom;
                delete prenom;
        }

};
```

Exercice 7

```
//chaine.h

class CChaine
{
        friend void affich(CChaine);
        char string [255];
        int n;

public:
        CChaine();
        CChaine(char*);
        virtual ~CChaine();

        bool operator ==(CChaine s);
        CChaine operator <<(CChaine s);

        void Majuscule();

};

////////////////////////////////////
// Chaine.cpp

#include "Chaine.h"
```

```
#include <iostream.h>
#include <string.h>
#include <ctype.h>
//Pour utiliser la fonction toupper
//toupper est une fonction C++ qui permet de changer la lettre en argument en majuscule

CChaine::CChaine()
{

}
CChaine::CChaine(char* s)
{
	strcpy(string,s);

}

CChaine::~CChaine()
{

}
bool CChaine::operator ==(CChaine s)
{
	if (strcmp(string,s.string)==0)
		return true;
	else return false;
}
CChaine CChaine::operator <<(CChaine s)
{
	CChaine t;
	strcpy(t.string,strcat(string, s.string));
	return t;
}

void CChaine::Majuscule()
{
	//convertir la chaine en majuscule
	for (int i=0;i<255; i++)
		toupper(string[i]);
}
```

Exercice 7

```
//vehicule.h
#include <iostream.h>
class Vehicule
{
	char* serie,*numero;
public:
	char* marque, *modele;
private:
```

```
        int puissance, jour, mois,annee, couleur;
public:
        int type;//1 pour Gazoil, 2 pour Essence

public:

        Vehicule()
        {
        }
        Vehicule (char* s,char* n) //constructeur de véhicule
        {
                serie =s;
                numero =n;

        }
        Vehicule (char*s,char* n,char* m,char* d)
        {
                serie =s;
                numero =n;
                modele =m;
                modele =d;
        }
        void Changer_date(int j,int m,int a)
        {
                jour=j;
                mois=m;
                annee=a;
        }
        void Changer_puissance(int p)
        {
                puissance=p;

        }
        void Changer_couleur (int c)
        {
                couleur =c;
        }
        void Affiche_caracteristiques()
        {
                cout <<"Modèle:"<<modele;
                cout <<"Marque:"<<marque;
                cout <<"numéro:"<<numero;
                cout <<"Série:"<<serie;
                cout <<"Date:"<<jour<<"/"<<mois<<"/"<<annee;

        }

public:

        virtual ~Vehicule()
```

```
        {
        }

};

//personne.h

#define NMAX 10
#include "vehicule.h" //Ne pas oublier le fichier de déclaration de Véhicule!
class Personne
{
        char* nom, *prenom;
public:
        int numero;
        char* rue_av, *ville;
private:

        Vehicule tvoiture[NMAX];
        int nb_voitures;

public:

Personne ();
Personne(char* nom);
Personne(char* nom,char* prenom);
void Changer_adresse (char* nville, int nnumero, char* nrue);
void Acheter_voiture(Vehicule v);
void Affiche_Homme();

public:

        virtual ~Personne();

};

//////////////////////////////////////////////////////

//Personne.cpp
#include "Personne1.h"

Personne::Personne()
{

        nb_voitures=0;
        prenom="foulen";
        nom="ben foulen";
}

Personne::~Personne()
{
```

```
}

Personne::Personne(char* n)
{
	nb_voitures=0;
	prenom="foulen";
	nom=n;

}
Personne::Personne(char* n,char* p)
{
	nb_voitures=0;
	prenom=p;
	nom=n;
}
void Personne::Changer_adresse (char* nville, int nnumero, char* nrue)
{
	ville= nville;
	numero=nnumero;
	rue_av=nrue;
}
void Personne::Acheter_voiture(Vehicule v)
{
	tvoiture[nb_voitures]=v;
	nb_voitures++;

}
void Personne::Affiche_Homme()
{
	cout <<"Je suis "<< prenom<< " "<<nom<<"\n";
	cout <<"Adresse:"<<numero<<","<<rue_av<<","<<ville<<"\n";
	cout <<"J'ai "<<nb_voitures<<" voitures\n";
	for (int i=0; i<nb_voitures;i++)

		tvoiture[i].Affiche_caracteristiques();
}

////////////////////////////////////////////////////////////
//Programme principal

void main()
{
	Personne p("Joulak", "Mondher");
	p.Changer_adresse("tunis", 5, "rue des oliviers");

	Vehicule v1("tunis 85", "1896", "Mercedes", "Classe E");
	v1.Changer_puissance(8);

	Vehicule v2("tunis 95", "489", "Peugeot", "Partner");
```

```
        v2.Changer_puissance(6);

        Vehicule v3("tunis 102", "6200", "Ford", "Fiesta");
        v3.Changer_puissance(4);

        p.Acheter_voiture(v1);
        p.Acheter_voiture(v2);
        p.Acheter_voiture(v3);

        p.Affiche_Homme();

}
```