



Exercice 3

- 1 - Proposer une spécification et une implémentation d'une classe *Personne* comportant trois attributs : *Nom*, *Prenom* et *Date_de_naissance*. La classe doit comporter :
 - Des constructeurs permettant de créer trois (3) objet de type *Personne* de manières différentes.
 - Un destructeur.
 - Deux méthodes de saisie et d'affichage des informations concernant une *Personne*.
 - une méthode *comparer* qui compare la date de naissance de deux personnes
- 2 - Proposer une fonction principale montrant les possibilités d'utilisation de cette classe.



EX3 Corrigé

```
class Personne
```

```
{
```

```
    Private :
```

```
        char Nom[20];
```

```
        char Prenom[20];
```

```
        Date Dn;
```

```
    public :
```

```
        Personne();
```

```
        Personne(char*, char*, Date);
```

```
        Personne(char*, char*, int, int, int);
```

```
        int comparer(Personne);
```

```
        ~Personne();
```

```
        void Saisir();
```

```
        void Afficher();
```

```
};
```



Corrigé

```
Personne::Personne(){}  
Personne::Personne(char* N, char* P, Date D)  
{  
    strcpy(Nom,N); strcpy(Prenom,P);  
    Dn = D;  
}
```

```
Personne::Personne(char* N, char* P, int J, int M, int A)  
{  
    strcpy(Nom,N); strcpy(Prenom,P);  
    Dn.jour=J; Dn.mois=M; Dn.annee=A;  
}
```

```
Personne::~~Personne(){} ENISO 2021-2022
```



Corrigé

```
Personne::Personne(){};
```

```
Personne::Personne(char* N, char* P, Date D) : Dn(D)
```

```
{
```

```
strcpy(Nom,N);
```

```
strcpy(Prenom,P);
```

```
}
```

```
Personne::Personne(char* N, char* P, int J, int M, int A) : Dn(J,M,A)
```

```
{
```

```
strcpy(Nom,N);
```

```
strcpy(Prenom,P);
```

```
}
```

```
Personne::~~Personne(){}; ENISO 2021-2022
```



Corrigé

```
void main()
{
    Personne P1;
    P1.Saisir();
    P1.Afficher();
    Personne P2("Ali", "Salah", Date(10,05,1999));
    P2.Afficher();
    Personne P3("Kamel", "Ramzi", 9,1,2000);
    P3.Afficher();
    if (P1.comparer(P2)==1)
        P1.afficher();
    else P2.afficher();
    system("PAUSE");
}
```

```
void Personne::Saisir()
{
    cout<<"\n Donner le nom : ";
    cin>>Nom;
    cout<<"\n Donner le prénom : ";
    cin>>Prenom;
    Dn.Saisir();
}

void Personne::Afficher()
{ cout<<"\n Nom : "<<Nom;
  cout<<"\n Prénom : "<<Prenom;
  Dn.Afficher();
}

int Personne::comparer(personne P1)
{  if (Dn.comparer(P1.Dn)==1)
    return (1);
    else return(2);
}
```



Exercice 4

1 - Proposer une spécification et une implémentation d'une classe *TabPersonnes* qui permet de gérer un groupe de personnes. Cette classe doit comporter :

- Un constructeur paramétré (N : nombre de personnes) permettant de créer un objet de type *TabPersonnes*.
- Un destructeur.
- Deux méthodes de saisie et d'affichage des informations concernant un ensemble de personnes.
- Une méthode de tri d'un ensemble de personnes suivant la date de naissance. Le tri se fait de la personne la plus jeune vers celle la plus âgée.

2 - Proposer une fonction principale montrant les possibilités d'utilisation de cette classe.



EX4 Corrigé

```
class TabPersonnes
{
    int Nb;
    Personne* Tab;
public :
    TabPersonnes(int);
    ~TabPersonnes();
    void Saisir();
    void Afficher();
    void Trier();
};
```



Corrigé

```
TabPersonnes::TabPersonnes(int N)
{ Nb = N;
  Tab = new Personne[Nb]; }
```

```
TabPersonnes::~~TabPersonnes() {delete[] Tab;}
```

```
void TabPersonnes::Saisir()
{for(int i = 0; i<Nb;i++)
  (Tab+i)->Saisir(); } //ou Tab[i].saisir();
```

```
void TabPersonnes::Afficher()
{for(int i = 0; i<Nb;i++)
  (Tab+i)->Afficher(); }
```




Corrigé

```
void main()
{ int Nombre;
  cout<<"Donnez le nombre de
  personnes : ";
  cin>>Nombre;
  TabPersonnes TP(Nombre);
  TP.Saisir();
  TP.Trier();
  TP.Afficher();
}
```

```
void TabPersonnes::Trier()
{ Personne Temp;
  int i,j, min;

  for(i=0;i<Nb-1;i++)
  { min = i;
    for(j=i+1;j<Nb;j++)
      if (Tab[min].Comparer(Tab[j])==1)
        min = j;
    if(min!=i)
    { Temp = Tab[i];
      Tab[i] = Tab[min];
      Tab[min] = Temp;
    }
  }
}
```



Exercice 5

Soit la classe Vecteur :

```
class vecteur {  
private:  
    int taille ;  
    float *vect;  
public:  
    vecteur(int N) ;  
    void saisir() ;  
    vecteur addition(vecteur V1);    //addition de deux vecteurs  
    float somme() ;                  // somme des éléments du vecteur  
    ~vecteur() ;  
}
```

A- Développer les fonctions de la classe ?

B- Proposer une implémentation avec la fonction main() ?



Exercice 5: corrigé

```
Void vecteur::vecteur(int N) {  
    taille = N;  
    vect = new [taille];  
}
```

```
void vecteur::saisie() {  
    int i;  
    for (i=0;i<N;i++)  
        cin>>vect[i];  
}
```

```
void vecteur::afficher()  
{  
    int i;  
    For (i=0;i<N;i++)  
        cout<<vect[i];  
}
```

```
Void vecteur::~~vecteur()  
{  
    delete [] vect;  
}
```



Exercice 5: corrigé

```
float vecteur::somme()  
{  
    float s=0.0;  
    for (i=0;taille; i++)  
        s += vect[i];  
    return s; }
```

```
vecteur vecteur::addition(float *V)  
{  
    float R = new[taille];  
    for (i=0;taille; i++)  
        R[i] = V[i] + *(this+i);  
    return R; }
```

```
main()  
{  
    vecteur V1, V2;  
    int N;  
    cout << « taille = ? »;  
    cin >> N;  
    V1.saisie(N);  
    V2.saisie(N);  
    cout << somme(V1);  
    (V2.addition(V1)).afficher();  
}
```