

Corrigé type d'examen du module Programmation C++ embarqué (durée : 1h30)

Spécialités : Electronique des systèmes embarqués.

ExerciceN°01 (08 points)

1- fonction '*rempl*' qui remplit le tableau 'Tab' par des valeurs entières entrées par l'utilisateur. (03pt)

```
// fonction qui remplit le tableau
void rempl (int *p, int taille)
{
    for (int*x=p; p<x+taille; p++)
    {
        cout << "Tab["<<p-x<<"]="=? ";
        cin >> *p;
    }
}
```

2- fonction '*somme*' qui retourne la somme des éléments de tableau 'Tab' (02pt)

```
/* fonction qui calcul la somme des
éléments du tableau*/
float somme (int *p, int taille)
{
    float somme=0;
    for (int*x=p; p<x+taille; p++)
    {
        somme+= *p;
    }
    return somme;
}
```

3- fonction '*ordre*' permet de trier le tableau 'Tab' selon la valeur de *O* (*O* = 'c': ordre croissant, *O* = 'd' : ordre décroissant) (03pt).

```
/* fonction ordre */
float ordre (int *pi, int taille, char o )
{ float aide=0;
  for (int*x=pi; pi<x+taille-1; pi++)
  {
    for (int *pj=pi+1; pj<x+taille; pj++)
    {
      if (o=='c')
      {
        if(*pj < *pi) {aide=*pi; *pi=*pj;*pj=aide;}
      }
      else if (o=='d')
      {
        if(*pj > *pi) {aide=*pi; *pi=*pj;*pj=aide;}
      }
    }
  }
}
```

ExerciceN°02 (12 points)

1. Deux attribusés 'x' et 'y' représentent le centre du cercle et deux autres 'r' et 'c' représentent respectivement le rayon et la couleur du cercle (02pt).

```
//attribués
private:
float x,y,r; char c;
```

2. Constructeur qui crée les objets de type 'cercle' et les initialise par les valeurs : $x=1.5$, $y=1.5$, $r=2$, $c='g'$ (02pt).

```
//Constructeur
cercle(): x(1.5),y(1.5),r(2),c('g') {}
```

3. Une méthode constante 'surf' qui retourne la surface du cercle (02pt).

```
// fonction qui calcul la surface
float surf ()
{
return 3.14*r*r;
}
```

4. Une méthode constante 'affiche' qui affiche les valeurs des attribusés 'x', 'y', 'r' et 'c' (02pt).

```
// fonction affiche()
float affiche ()
{
cout << " centre =" << x << ", " << y << endl;
cout << " rayon =" << r << endl;
cout << " couleur =" << c << endl;
}
```

5. Une méthode 'modif_r' qui modifie le rayon du cercle par une valeur précisée par l'utilisateur (elle n'accepte pas les valeurs >10 ou les valeurs <1) (02pt).

```
// fonction qui modifie le rayon
void modif_r ()
{
do
{
cout << " Entrer le nouveau rayon ? ";
cin >> r;
}
while (r >10 || r <1 );
}
```

6. Une méthode 'modif_c' qui modifie la couleur du cercle par une valeur précisée par l'utilisateur (elle n'accepte que les valeurs 'g', 'y' ou 'b') (02pt).

```
void modif_c ()
{
do
{
cout << " Entrer la nouvelle couleur (g, y ou b) ? ";
cin >> c;
}
while (c != 'g' && c != 'y' && c != 'b' );
}
```

7. Une opérateur surchargé '>' qui compare les objets de la classe cercle selon leur surfaces (choisissez **bool** comme type de retour) (02pt).

```
//Surchage de l'opérateur '>'  
bool operator > ( cercle c1 )  
{  
    if (3.14*r*r > c1.surf() )  
        return true;  
    else  
        return false;  
}
```