

Corrigé de l'Epreuve de Moyenne Durée

Exercice : 1 (7pts)

1-(3 pts)

```
#include <iostream>
using namespace std ;

int diviser (int exp, int var)
{
    int r;
    r= exp*var ;
    return r;
}

main( )
{
    int m;
    int n = 12 ;
    int p = 3 ;
    m=diviser (2*n, n+p) ;
    cout << " m = " << m << "\n" ;
}
```

2-(4pts)

a) 2pts

```
#include <iostream>
using namespace std;
int main()
{ float t1[10] ;
  int som,i;
  som=0;
  for ( i=0;i<9;i++) {cin>>*(t1+i); }
  for (i=0;i<9;i++) { som=som+*(t1+i);
  for (i=0;i<9;i++) cout<<som<<endl;
  return 0; }
```

b) 2 pts

```
#include <iostream>
using namespace std;
int main()
{ float t1[10] ;
  int som,i;
  som=0;
  for ( i=0;i<9;i++) {cin>>t1[i]; }
  for (i=0;i<9;i++) { som=som+t1[i];
  for (i=0;i<9;i++) cout<<som<<endl;
  return 0; }
b)
```

Exercice 2 : 6pts

```
#include <iostream.h>
                2 pts /* fichier vecteur.H */
/* déclaration de la classe point */
class vecteur
{
private :
double x, y, z ;
public :
vecteur () ; // constructeur 1
vecteur (double, double, double) ; // constructeur 2
void affiche () ;
};

                fichier vecteur .cpp. 2pts
```

```
/* déclaration de la classe point */
vecteur::vecteur ()
{x=0 ; y=0 ; z=0 ;}
vecteur::vecteur (double a, double b, double c)
{x = a ; y = b ; z = c ;}
```

```
void vecteur::affiche ()
{cout << "Vecteur de coordonnées : " << x << " "
<< y << " " << z << "\n" ;}
```

/* programme de test de la classe vecteur */ **2 pts**

main()

```
{
vecteur v1 (1,2,3) ;
vecteur v2 (5,4,3) ;
vecteur v3 ;
v1.affiche () ; v2.affiche () ; v3.affiche () ;
}
```

Exercice 3 : (6 pts)

1)
/* fichier POINT.H */ **2.5pts**

```
/* déclaration de la classe point */  
class point  
{  
  
float x, y ; // coordonnées (cartésiennes) du point  
public :  
point (float, float) ; // constructeur  
void deplace (float, float) ; // déplacement  
void homothetie (float) ; // homothétie  
float abscisse () ; // abscisse du point  
float ordonnee () ; // ordonnée du point  
};  
2)
```

/* fichier POINT.H */ **2.5 pts**

```
#include "point.h"  
#include <cmath> // pour sqrt et atan  
#include <iostream>  
using namespace std ;  
const float pi = 3.141592653 ; // valeur de pi  
float angle (float, float) ; // fonction de service (non  
membre)  
/***** définition des fonctions membre  
*****/  
point::point (float abs, float ord)  
{ x = abs ; y = ord ;  
}  
void point::deplace (float dx, float dy)  
{ x += dx ; y += dy ;  
}  
void point::homothetie (float hm)  
{ x *= hm ; y *= hm ;  
}  
float point::abscisse ()  
{ return x ;  
}  
float point::ordonnee ()  
{ return y ;  
}
```

3) Programme d'essai: 2 pts

```
#include <iostream>  
using namespace std ;  
#include "point.h"  
main ()  
{  
point p (1.25, 2.5) ; // construction  
// affichage  
cout << "Coordonnées cartésiennes : " << p.abscisse  
() << " "  
<< p.ordonnee () << "\n" ;  
p.deplace (2.1, 3.4) ; // déplacement  
// affichage  
cout << "Coordonnées cartésiennes : " << p.abscisse  
() << " "  
<< p.ordonnee () << "\n" ;  
}
```