Exercice 1 (07 points)

**Considérons l'algorithme ACO pour résoudre le TSP symétrique avec n=4 villes A, B, C, D telles que : AB=5, AC=2, AD=1, BC=2, BD=5, CD=4 et la configuration de paramètres suivante :**

**Q=12, α=1, β=1 , τ0=1 ; ρ = 0.25.**

**Supposons qu'après la première itération de l'algorithme, chaque fourmi ait construit l'un des deux tours ABCD et ABDC.**

1) **Calculer la longueur de chacun de ces 2 tours.**
2) **Calculer la quantité de phéromone après la première itération sur chaque arrête.**
3) **Supposons qu'on dispose d'une fonction Tour(k) qui retourne le tour (la solution, le cycle hamiltonien) de la fourmi k et d'une fonction Length(x) qui retourne la longueur d'un tour x.**
   a) **Ecrire l'algorithme de mise à jour de phéromone avec m fourmis pour un TSP de n villes. (Utiliser ces deux fonctions sans les implémenter.)**
   b) **Evaluer la complexité de cet algorithme.**

Exercice 2 (05 points)

**L'algorithme A ci-contre permet de résoudre un problème d'optimisation combinatoire.**

1) **En relever la formulation mathématique de ce problème, l'espace de recherche et sa taille.**
2) **Qu'appelle-t-on cette méthode et quels sont ses paramètres (noms et valeurs).**
3) **Evaluer la complexité de cet algorithme.**
4) **Que fait exactement cet algorithme ?**

```
void Algo() {
int F(int x)) { return abs(x*x*x+ 2*x-100) ;}
g = 0 ;
for ( i=0; i<m ; i++)
      {   x[i]  = 1+int(50*random(0,1));
          v[i]  = 1+int(5*random(0,1));
          p[i]  = x[i]  ;
          if (F(p[i]) > F(g))  g = p[i] ;
      }
for (iter =0; iter < MaxIter ; iter++)
      for ( i=0; i<m ; i++)
           {  v[i]=0.9*v[i]+ 0.5*abs(x[i]- p[i])+0.5*abs(x[i]- g) ;
              x[i]= x[i] +  v[i] ;
              if ( x[i] >50) x[i] =50 ;
              if (F(x[i]) > F(p[i]) )
                   { p[i] = x[i];
                     if (F(p[i]) > F(g))   g = p[i] ;
                   }
           }
  printf (g , F(g)):  }
```

Exercice 3 (08 points)

**On se propose de maximiser la fonction $f$ à deux variables entières x, y telle que :**
**$f(x,y) = x^2 + y^2$ - 20 en utilisant un algorithme génétique avec une population de 4 individus.**

1) **Calculer les probabilités de sélection par rang de chacun des individus :**
   **x1 = (6,3) ; x2 = (5,5) ; x3 = (6,0) ; x4 = (0,7).**
2) **Considérons qu'on utilise une population de m individus.**
   a) **Ecrire l'algorithme d'une fonction entière *Rank(x)* qui retourne le rang d'une solution x dans la population.**
   b) **Utiliser l'algorithme de (a) pour élaborer un algorithme d'une fonction entière *Selected()* qui retourne l'indice de l'individu sélectionné par rang de la population.**
   c) **Utiliser l'algorithme de (b) pour élaborer un algorithme de sélection par rang pour toute la population.**