ROYAUME DU MAROC
Ministère de l'Education Nationale, de la Formation Professionnelle, de l'Enseignement Supérieur et de la Recherche Scientifique
Secrétariat d'Etat Chargé de l'Enseignement Supérieur et de la Recherche Scientifique

de Casablanca Université Hassan II
Ecole Normale Supérieure

de l'Enseignement Technique Mohammedia

جـامعـة الحسن الثاني بالدار البيضاء
UNIVERSITÉ HASSAN II DE CASABLANCA

المملكة المغربية

وزارة التربية الوطنية و التكوين المهني والبحث العلمي والتعليم العالي

كتابة الدولة المكلفة

بالتعليم العالي والبحث العلمي

جامعة الحسن الثاني بالدار البيضاء

المدرسة العليا لأساتذة التعليم التقني

المحمدية

Concours d'accès en première année du cycle d'ingénieurs pour les filières :

- **Génie du Logiciel et des Systèmes Informatiques Distribués (*GLSID)***
- **Ingénierie Informatique, Big Data et Cloud Computing (II-BDCC)**

Session : Juillet 2017
Epreuve d'Informatique

Durée : 3 heures

Remarques importantes :

- L'épreuve se compose de trois parties :
  - Partie Informatique générale (temps recommandé : 30 mn), **(Notée sur 20 points)**
  - Partie Algorithmique et Programmation (temps recommandé : 2h), **(Notée sur 80 points)**
  - Partie techniques d'expression (temps recommandé : 30 mn), **(Notée sur 20 points)**
- L'usage de la calculatrice ou de tout autre appareil électronique est interdit.
- Aucun document n'est autorisé.
- **Pour la partie « Informatique générale » :**
  - Les réponses aux questions doivent être reportées sur la grille de réponse fournie avec cette partie
- **Pour la partie « Algorithmique et programmation » :**
  - Les réponses à toutes les questions doivent être rédigées dans les feuilles de réponses.
  - Chaque question du QCM ne peut avoir qu'une seule réponse possible parmi les quatre choix dont la lettre correspondante (A, B, C ou D) est à reporter dans les feuilles de réponses.
  - Les solutions algorithmiques peuvent être rédigées en utilisant un pseudo langage algorithmique ou l'un des langages de programmation suivants : C, C++, C#, Java
  - La clarté et la précision de votre solution algorithmique sera prise en considération.
- **Pour la partie « Techniques d'expression » :**
  - La rédaction de la réponse doit être rendue dans la feuille fournie.
  - La réponse à cette partie est obligatoire.
- Aucune autre explication supplémentaire ne sera fournie aux candidats en cours de l'examen.

Partie 2 : Algorithmique et Programmation (2 H)

- Les réponses à toutes les questions doivent être rédigées dans les feuilles de réponses.
- Chaque question du QCM ne peut avoir qu'une seule réponse possible parmi les quatre choix dont la lettre correspondante (A, B, C ou D) est à reporter dans les feuilles de réponses.
- Les solutions algorithmiques peuvent être rédigées en utilisant un pseudo langage algorithmique ou l'un des langages de programmation suivants : C, C++, C#, Java
- La clarté et la précision de votre solution algorithmique sera prise en considération.

QCM (25 points) :

Reportez dans les feuilles de réponses, la lettre qui correspond à la bonne réponse pour les questions suivantes :

1. On considère la fonction suivante :

```
int fonction1(){
 int i,n; i=0;  n=0;
 int T[]={1,5,8,4,6,7,44,10,20,2};
 while(true){
      if(n==5) break;// break permet de sortir de la boucle
            n=n+1;
            if(i>9) i=i-4;
            if(i<0) i=i+3;
            if (T[i]<T[i+1]) i=i+3;
            else
            i=i-4;
}
  return T[i];
}
```

L'appel de cette fonction permet de retourner :

A) 8 B) 4

C) 6 D) 20

2. Soit la fonction suivante :

```
int fonction2(int n) {
  int a;  int b;  int c;   int i;
   a = 0;   b = 1;
   if (n <= 1) {
      return n;
      }
   else {
    i = 1;
    while (i < n) {
      c = a + b;
      a = b;
      b = c;
      i = i + 1;
       }
  }
 return c;
}
```

Pour n=9, la valeur retournée par fonction2 est :

A) 9
B) 10
C) 34
D) 35

3. Soit la fonction suivante :

```
int fonction3(){
      int a[]={10,20,-35,35,-10,5,-10,20,30};
      int n=9;int  s1 = 0; int s2 = 0;
      for( int i = 0, j = 0; j < n; j++ ){
            s1 += a[ j ];
            if( s1 > s2 ) {
              s2 = s1;
            }
            else if( s1 < 0 ) {
              i = j + 1;
              s1 = 0;
            }
      }
 return s2;
}
```

L'appel de cette fonction permet de retourner :

A) 30
B) 65
C) 55
D) 70

4. Soit la fonction récursive suivante :

```
void fonction4(int n){
      if(n>0){
          fonction4(n-1);
          printf("%d ",n);
          fonction4(n-1);
      }
}
```

Pour n=4, la fonction permet d'afficher :

A) 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1 B) 4 3 2 1 1 2 3 4

C) 4 3 2 1 1 2 3 4 3 2 1 1 2 2 1 D) 1 2 3 1 2 3 4 4 3 2 1 3 2 1

5. Quelle est la complexité dans le pire cas de la recherche d'un élément dans un arbre binaire de recherche de hauteur h contenant n nœuds ?

A) Θ(n) B) Θ (h)

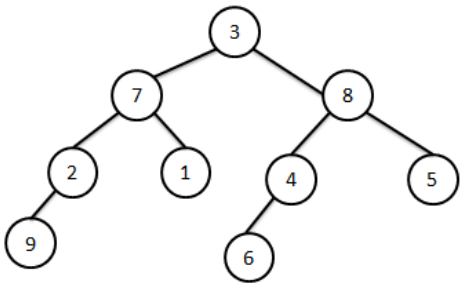
C) Θ (log n) D) Θ (log h)

6. On insère les éléments 4, 3, 12, 7, 9 (dans cet ordre) dans un *tas*. Dans quel ordre vont-ils ressortir ?

A) 9, 7, 12, 3, 4 B) 3, 4, 7, 9, 12

C) 4, 3, 12, 7, 9 D) 12, 9, 7, 4, 3

7. Soit l'arbre binaire A suivant :




```
void afficherContenuArbre(Noeud *A){
// la structure de l'arbre, la structure de la pile et les //opérations
sur la pile sont considérés déjà définis
     Noeud *pt;  Element *pil; // pil est une pile
     pil=NULL;// initiliser la pile à null
     pt=A;      //la pile reçoit la racine de l'arbre
  while(pt||pil){
     if(pt){
       Empiler(pt,&pil); //procedure qui empile pt à pil
       pt=pt->FG;
     }
     else{
       afficherTetePile(pil); // procédure qui affiche le
                              //contenu de la tête de la pile
```

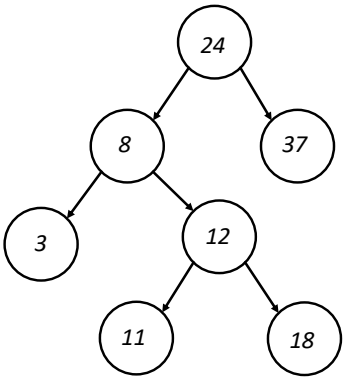
```
            pt=pil->contenu->FD;
            Depiler(&pil);//procédure qui dépile un élément de la pile
          }
      }
  }
```

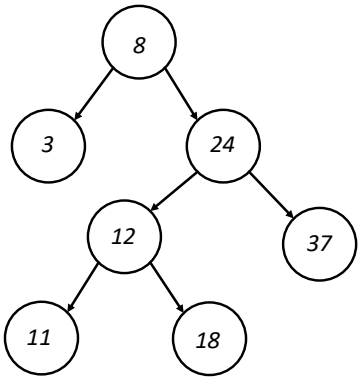
L'appel à cette fonction, avec le paramètre l'arbre de la figure ci-dessus, permet d'afficher :

A) 9 2 7 1 6 4 8 5 3
B) 3 7 8 2 1 4 5 9 6
C) 9 2 1 7 6 4 5 8 3
D) 9 2 7 1 3 6 4 8 5

8. Quelle opération transforme l'arbre de gauche de la figure ci dessous en celui de droite ?

A) une rotation droite,
B) une double rotation,
C) une rotation gauche,
D) aucun des trois.





9. A laquelle des structures suivantes s'apparente le plus une représentation de graphe par listes de successeurs ?

A) une pile,
B) un arbre binaire,
C) une table de hachage,
D) un tableau bidimensionnel.

Exercice 1 (5 points) :

Soit l'expression suivante :

$$X=1+(1+2)*2+(1+2+3)*2^2+(1+2+3+4)*2^3+ \ldots +(1+2+3+4+\ldots +n)*2^{n-1}$$

1. Ecrire la forme itérative de l'algorithme qui permet de calculer X
2. Ecrire la forme récursive de l'algorithme qui permet de calculer X

Exercice 2 (20 points) :

ESP est un espace d'une seule dimension qui contient un ensemble de points P(x) où x représente l'abscisse du point P. Dans notre cas, on suppose que ESP est formé des points : A(1), B(3), C(5), D(9), E(11), F(7) et G(4).

L'algorithme KM, ci-dessous, permet de regrouper ces points en deux **groupes homogènes** G1 et G2.

1. **Choisir de manière aléatoire 2 points formant chacun un groupe. Les groupes formés G_1 et G_2 ont respectivement les centres C1 et C2 (un centre représente la moyenne des points d'un groupe)**
2. **Affecter chaque Point X au groupe G_i ayant le centre le plus proche**
3. **Recalculer le centre Ci de chaque groupe en utilisant la moyenne des points le composant**
4. **Aller à l'étape 2 si la condition d'arrêt n'est pas vérifiée (la condition n'est pas vérifiée au moment où aucun changement ne peut y avoir lieu après deux itérations successives).**

Questions :

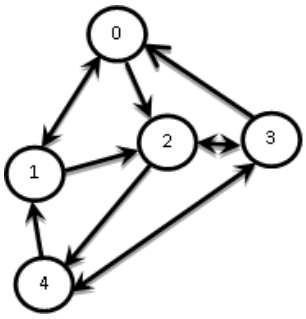
1) L'exécution de la première instruction de l'algorithme KM en choisissant les points A et B donne lieu aux groupes G1 et G_2 ayant respectivement les centres C_1=A(1) et C_2=B(3). Ce résultat peut être représenté de la manière suivante:

Compléter, à partir de l'instruction 2, l'exécution de l'algorithme KM sur l'ensemble des points de l'espace ESP pour former les deux groupes G_1 et G_2 tout en schématisant les groupes obtenus pour chaque itération.

2) Proposer un algorithme qui permet de regrouper, d'une manière générale, les n points d'un espace ESP en k groupes.

Exercice 3 (30 points) :

On considère un réseau routier représenté par un graphe orienté. Les nœuds du graphe représentent des villes 1, 2, … n. Les arcs du graphe représentent les routes reliant les villes. Une route peut être soit unidirectionnelle ou bidirectionnelle. Un chemin entre les nœuds i et j mesure une distance d (i, j). La figure suivante représente un exemple de graphe correspondant à un réseau routier reliant 5 villes.



Nous souhaitons représenter ce graphe par une matrice d'interconnexion carrée M de dimension (n, n) ou n représente le nombre de nœuds. La valeur de M (i , j) représente la distance entre le nœud i et le nœud j si un chemin existe du nœud i vers le nœud j. Si non, la valeur de M (i, j) est représentée par -1. La matrice suivante montre un exemple représentant le cas du graphe précédent.

	0	1	2	3	4
0	0	5	4	-1	-1
1	5	0	3	-1	-1
2	-1	-1	0	2	4
3	5	-1	2	0	6
4	-1	2	-1	6	0

Chaque ville est caractérisée par son nom et ses coordonnées géographiques (latitude, longitude et altitude) et le nombre de population. Les villes sont stockées dans un tableau NOEUDS[n].

1. Ecrire le code qui permet de déclarer les structures représentant une ville et un réseau routier en utilisant des types structurés, des tableaux ou des classes.
2. Ecrire le code d'une fonction qui permet de retourner la distance linéaire entre un nœud i et un nœud j en utilisant les coordonnées géographiques de chaque nœud.
3. Ecrire le code qui permet de retourner la longueur d'un chemin traversant une liste des nœuds.
4. Ecrire le code d'une fonction qui permet de déterminer les différents chemins possibles entre un nœud i vers un nœud j.
5. Ecrire le code d'une fonction qui permet déterminer le chemin le plus proche d'un nœud i vers un nœud j.
6. En plus de la distance, nous souhaitons aussi prendre en considération d'autres caractéristiques des chemins tels que la vitesse maximale à ne pas dépasser, le débit instantané du chemin (Nombre de véhicules par seconde), existence ou nom d'une station de service, etc …
    A) Proposer une déclaration de la structure représentant le nouveau réseau routier.
    B) Expliquer les contraintes qui vont peser sur la détermination du chemin le plus rapide allant d'un nœud i vers un nœud j.