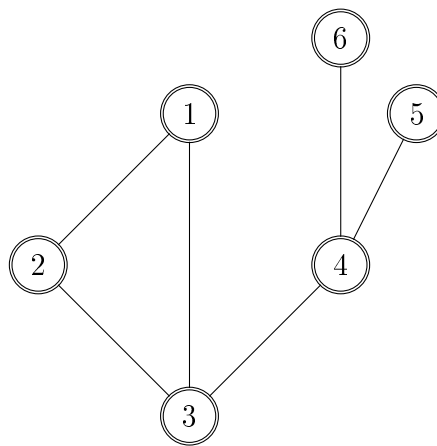




Institut Supérieur d'Informatique et de Multimédia de Sfax
Année Universitaire: 2014-2015
Module: Théorie des graphes et optimisation
Auditoire: LAMW 3
Enseignants: Mohamed TOUNSI & Wided ZENATI
TP n°: 3

Exercice n° 1

1. Implémenter le graphe G ci-dessous. Utiliser la bibliothèque **NetworkX**.
2. Dessiner le graphe G. Utiliser la bibliothèque **PyLab**.



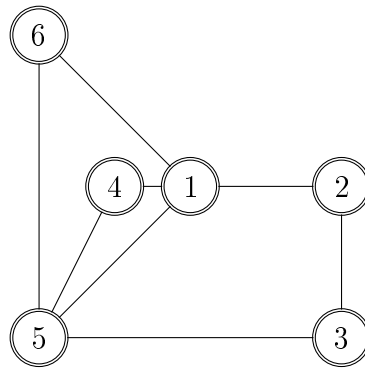
Graphe G

3. Quel est le nombre de sommets dans le graphe G ?
4. Quel est le nombre d'arêtes dans le graphe G ?
5. Enlever l'arête (3,4). Dessiner le graphe et vérifier le résultat obtenu.
6. Quels sont les composants connexes dans le nouveau graphe ?

Exercice n° 2

1. Écrire une fonction **verif_eulerien** qui prend une variable en entrée de type *Graphe* et retourne *True* si le graphe est eulérien et *False* sinon.
2. Créer deux graphes complets **A** (K_3) et **B** (K_4). Vérifier avec la fonction **verif_eulerien** si les graphes *A* et *B* sont eulérien ou non ?

3. Implémenter le graphe C ci-dessous. Le graphe est-il eulérien ?
4. Exploiter la méthode `eulerian_circuit` de `networkx` pour générer un circuit eulérien à partir du sommet 3.
5. Exploiter la méthode `all_simple_paths` de `networkx` pour générer tous les chemins qui commencent par le sommet 1 et terminent par le sommet 3.



Graphe C