Nom et prénom : .....

Devoir Surveillé		
Classe : *SEM31*	***Matière : D****éveloppement* ***M****obile* ***A****vancé*	/20

Exercice1 (10 points)

```
private void ajouter() {
```

.....

```
}
private void remplir() {
```

.....

```
}
```

```
protected void actualiser() {
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
}
private void modifier() {
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....


}
```

Exercice2 (10 points)

```
private void ajouter() {
  RequestQueue queue =
  String url =
  StringRequest req =
          new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
           try{
              ----------
              ----------
              ----------
              ----------
              ----------
              ----------
              ----------
              ----------
              ----------
              ----------
            } catch (JSONException error) {
            Toast.makeText(Ajout.this, "Problème d'analyse JSON: " +
                        error.getMessage(), Toast.LENGTH_LONG).show();
            }
            }
          }, new Response.ErrorListener() {
               @Override
               public void onErrorResponse(VolleyError e) {
            Toast.makeText(Ajout.this, "Problème d'appel HTTP: " +
                         e.getMessage(), Toast.LENGTH_LONG).show();
               }
             }) {
        @Override
        public Map<String, String> getParams() throws AuthFailureError {
          HashMap<String, String> headers = new HashMap<String, String>();
          ----------
          ----------
          ----------
          ----------
          return headers;
        }
        };
        ----------
}
```

```
private void rechercher() {
  RequestQueue queue =
  String url = ----------
  StringRequest sr =
          new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
               try{
                  ----------
                  }
     } catch (JSONException error) {
      Toast.makeText(Recherche.this, "Problème d'analyse JSON: " +
     error.getMessage(), Toast.LENGTH_LONG).show();
    }  }  }, new Response.ErrorListener() {
               @Override
            public void onErrorResponse(VolleyError e) {
     Toast.makeText(Recherche.this, "Problème d'appel HTTP: " +
     e.getMessage(), Toast.LENGTH_LONG).show();
               }  }) {
       @Override
       public Map<String, String> getParams() throws AuthFailureError {
         HashMap<String, String> headers = new HashMap<String, String>();
         ----------
         return headers;
       } };
        ----------

}
```