## Préparer les tests dans le cadre de la SAE S1.01

Vous allez devoir préparer les tests des fonctions/procédures de votre projet Puissance4. Deux possibilités s'offrent à vous : rédiger les différents cas de test par exemple à l'aide d'un fichier Excel comme vu lors de la séance sur les tests, ou bien préparer un fichier JSON qui sera ensuite analysé pour produire les résultats de tests automatiquement.

1 - Vous pouvez rédiger dans un classeur type Excel les différents cas de test, avec une feuille par fonction (voir le fichier "Cahier de test.ods" fourni en pièce jointe).

Données de test	Résultats attendus	Résultats obtenus	Résultat du test (réussi ou non).
	Données de test	Données de test Résultats attendus	Données de test Résultats attendus Résultats obtenus

2 – Ou bien vous pouvez préparer des tests automatiques en créant un fichier unique appelé tests. json. Ce fichier devra respecter la structure décrite dans l'exemple suivant.

Dans cet exemple, on prépare les tests de 2 fonctions (somme et diff) avec à chaque fois 2 cas de tests.

```
[
 {
                                                   numéro du cas de test
    "test num": 1,
    "test label": "Test sum nb pos & neg",
                                                              programme qui contient
    "file name": "somme.c", —
                                                        la fonction à tester (à écrire plus tard)
    "goal": {
      "type": "int",
      "val": "-3"
                                                     valeur attendue (goal)
    },
                                                          et son type
    "args": [
        "type": "int",
        "val": "4"
                                                        liste des valeurs du cas de test
                                                       (les "arguments") c'est-à-dire les
      },
                                                           paramètres nécessaires
                                                                à la fonction
        "type": "int",
        "val": "-7"
    1
  },
                                                                        NOTE
                                                           Pour le goal et pour les arguments,
    "test num": 2,
                                                                les types possibles sont :
    "test label": "Test sum nb neg & neg",
                                                                         int,
    "file name": "somme.c",
                                                                        char.
    "goal": {
                                                                        float,
      "type": "int",
                                                                        string,
      "val": "-11"
                                                                        grille,
    },
                                                                         bool.
    "args": [
      {
        "type": "int",
        "val": "-4"
      },
        "type": "int",
        "val": "-7"
      }
    ]
  },
  {
```

```
"test num": 3,
    "test label": "Test substraction nb pos & neg neg after the minus sign",
    "file name": "diff.c", ____
                                                     ici on teste une autre fonction
    "goal": {
     "type": "int",
     "val": "17"
    } ,
    "args": [
     {
       "type": "int",
       "val": "12"
     },
       "type": "int",
       "val": "-5"
      }
   ]
  },
    "test num": 4,
    "test label": "Test substraction nb both neg, with second absolute val
greater than first",
    "file_name": "diff.c",
    "goal": {
     "type": "int",
     "val": "7"
    },
    "args": [
     {
       "type": "int",
       "val": "-5"
      },
       "type": "int",
       "val": "-12"
    ]
  }
1
```

Après les vacances, ce fichier sera exploité par un programme qui vous sera fourni et qui exécutera automatiquement tous les tests qui y sont décrits.