



# Initiation à l'Algorithmique et à la Programmation

## Mémoire de projet





---

Page de Garde.....	1
Table des matières.....	2
Présentation de l'application.....	3
Organisation des tests.....	5
Bilan de validation des tests.....	7
Bilan de Projet.....	9
Annexe 1 : Jeux de test et sorties.....	11
Annexe 2 : Sources Sprint 5 Validé.....	14

---



## **Introduction :**

Pour travailler sur des données, on a souvent besoin de les classer, de les organiser, pour faciliter leur traitement par l'homme. L'utilisation d'un tableau est la plus appropriée, Une table de données est un ensemble de données organisé sous forme d'un tableau où les colonnes correspondent à des catégories d'information d'un type donné et les lignes à des enregistrements.

## **Problématique :**

Programmer un interpréteur de commandes pour gérer un tableau de données.

## **Rôle fonctionnel du projet :**

L'objectif de ce projet est de programmer un interpréteur de commande qui permet de créer (puis supprimer) une table de donnée en définissant son schéma d'organisation, c'est-à-dire les noms des données qu'elle va contenir ainsi que leur type. Il doit ensuite être possible d'afficher ce schéma, de créer des lignes dans le tableaux, d'en afficher, une, plusieurs ou certaines, ainsi que d'en supprimer. Il faut aussi pouvoir quitter le programme grâce à une commande.

## **Les entrée et sorties :**

Ce projet possède neuf entrées, qui correspondent aux neuf commandes à coder : `Create_table`, `Delete_table`, `Afficher_schema`, `Insert_enregistrement`, `Afficher_enregistrement`, `Afficher_enregistrements`, `Delete_enregistrement`, `Select_enregistrement`, `Exit`.

La reconnaissance de ces neuf commandes s'opère dans le main, par l'utilisation de la fonction `strcmp()` issue de la bibliothèque `<string.h>`, qui permet de comparer deux chaînes de caractère. Elle retourne 0 si les chaînes sont identiques et autre chose si elles sont différentes.



Son prototype est le suivant : `int strcmp(const char* str1, const char* str2);`  
On utilise donc son retour associé à une condition pour reconnaître les commandes, pour ensuite appeler la fonction qui correspond à chaque commande.

Pour ce qui est des sorties, seules les commandes d'affichage sont associées à des sorties. En effet, ces fonctions contiennent des commandes d'affichage `printf`. La fonction `Afficher_schema` affiche l'organisation d'un tableau, ce sont les noms et types de données choisis à la création de la table. De même les fonctions `Afficher_enregistrement` et `Afficher_enregistrements`, permettent d'afficher respectivement une ou toutes les lignes d'une table. Enfin, la commande `Select_enregistrement` permet d'afficher un intervalle précis de ligne de la table selon un critère précis.

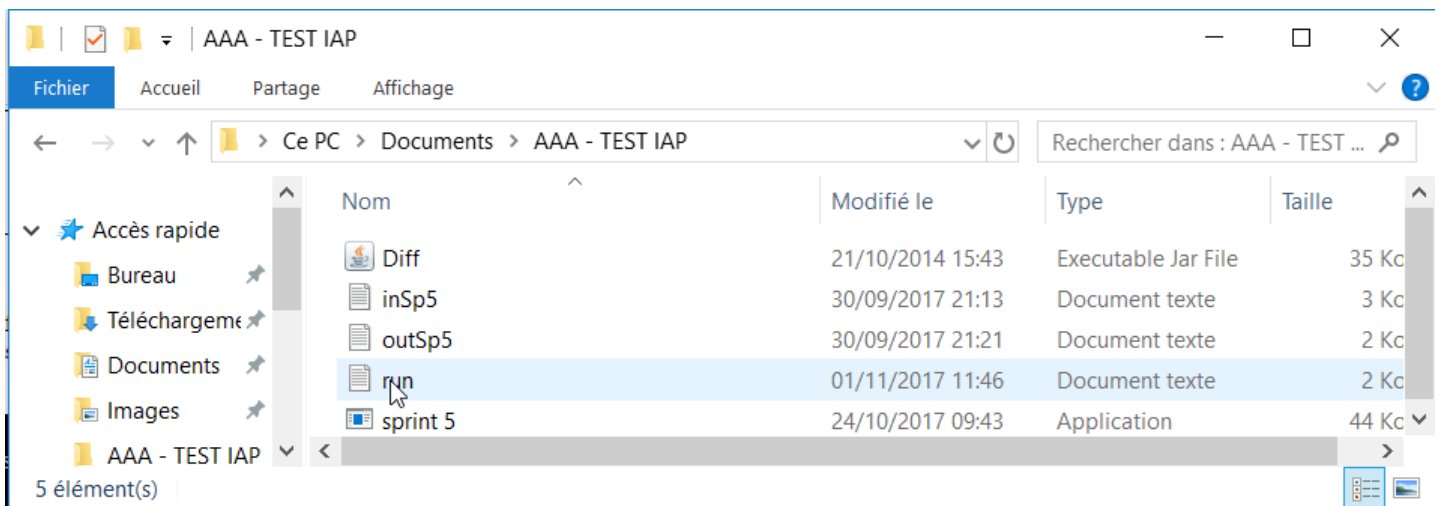
# II/ Organisation des tests



Le projet a été mené sous forme de sprints définis par le génie logiciel. Un sprint est défini par une spécification, un jeu de données de test (inSp#) ainsi que des résultats attendus (OutSp#).

Un test permet de valider un objectif. Il utilise donc le jeu de données de test (JDT) et le résultat précis. Nous avons donc utilisé la redirection de flux d'entrée pour tester rapidement et efficacement nos programmes afin de valider nos sprints.

Nous avons donc créé un dossier contenant les fichiers requis :



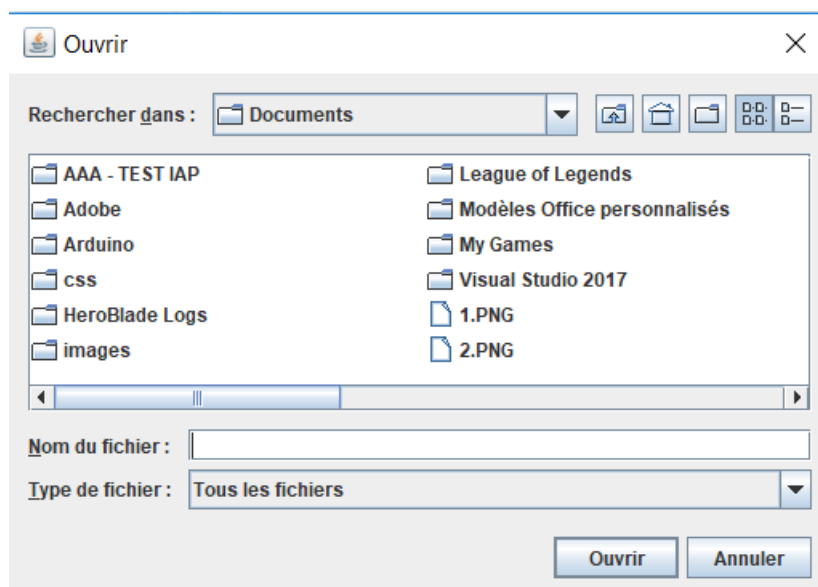
Utilisation de l'invite de commande pour la redirection :

```
Microsoft Windows [version 10.0.16299.19]
(c) 2017 Microsoft Corporation. Tous droits réservés.
C:\Users\arsen>cd Onedrive
C:\Users\arsen\OneDrive>cd Documents
C:\Users\arsen\OneDrive\Documents>cd "AAA - TEST IAP"
C:\Users\arsen\OneDrive\Documents\AAA - TEST IAP>"sprint 5.exe" <inSp5.txt>
run.txt
C:\Users\arsen\OneDrive\Documents\AAA - TEST IAP>
```

On crée ainsi un fichier run.txt qui contient les sorties de notre programme.



Il s'agit ensuite de comparer ce fichier de sorties run.txt avec le fichier des sorties attendues OutSp#.txt. Pour cela on utilise un outil fourni par les professeurs, diff.jar :



Ce programme nous montre ainsi les différences entre le fichier de sortie produit par notre programme et le fichier de sorties attendues. Si des différences apparaissent, il convient de revoir le programme, sinon, il est déclaré 0-défaut, et on passe à la suite.

# III/ Bilan de validation des tests



Pour le bilan de la validation de tests, notre projet valide les cinq sprints.

Pour le premier sprint, il faut créer les types structurés `Table` et `Champ`, ainsi que de coder les fonctions correspondant aux commandes `Create_table` et `Afficher_schema`. Pour `Create_table`, il faut enregistrer le nom et l'organisation de la table de données, c'est-à-dire le nombre des champs, leur nom et le type de données que chacun va contenir. Pour `Afficher_schema`, il faut afficher le nom de la table et le nombre de Champs qu'elle contient, les noms de Champs et types de données enregistrés précédemment, et cela sur plusieurs lignes. Elle affiche également « table inconnue » si la table demandée n'existe pas. La comparaison des fichiers obtenus par redirection nous a montré des espaces en trop à certains endroits, ce que nous avons facilement corrigé, nous sommes ensuite passés au sprint de niveau supérieur.

Pour le deuxième sprint, il faut créer le type structuré `Enregistrement` et `Data`, ainsi que coder les fonctions correspondant aux commandes `Insert_enregistrement` et `Afficher_enregistrements`. Pour `Insert_enregistrement`, il faut enregistrer une ligne de données contenant des données dans chaque champ correspondant. Pour `Afficher_enregistrements`, il faut afficher les lignes de données de la table. Elles affichent également « table inconnue » si la table demandée n'existe pas. La comparaison des fichiers obtenus par redirection nous a permis de déclarer ce sprint comme 0-défauts et de passer au sprint de niveau supérieur.

Pour le troisième sprint, il faut coder les fonction correspondant aux commandes `Delete_enregistrement` et `Afficher_enregistrement`. Pour `Delete_enregistrement`, il faut supprimer une ligne de données de la table. Pour `Afficher_enregistrement`, il faut afficher une ligne de données précise de la table. Elles affichent également « table inconnue » si la table demandée



n'existe pas. La comparaison des fichiers obtenus par redirection nous a permis de déclarer ce sprint comme 0-défaut et de passer au sprint de niveau supérieur.

Pour le quatrième sprint, il faut coder la fonction correspondant à `Delete_table`, il faut donc supprimer la table. Elle affiche également « table inconnue » si la table demandée n'existe pas. La comparaison des fichiers obtenus par redirection nous a permis de déclarer ce sprint comme 0-défaut et de passer au sprint de niveau supérieur.

Pour le cinquième et dernier sprint, il faut créer le type structuré `Date` et le type énuméré `Bool`. Il faut aussi coder les fonctions correspondant à `Select_enregistrement`. `Compare_enregistrement` détermine si une ligne de donnée fait partie de l'intervalle demandé et renvoie un booléen à `Select_enregistrement` qui affiche toutes les lignes de données de la table comprises dans l'intervalle. Elle affiche également « table inconnue » si la table demandée n'existe pas. La comparaison des fichiers obtenus par redirection nous a permis de déclarer ce sprint comme 0-défaut et de passer au sprint de niveau supérieur.

# IV/ Bilan du projet



Au cours de ce projet nous avons rencontré un certain nombre de problèmes, mais nous avons pu y apporter les solutions adéquates.

Tout d'abord, nous nous sommes heurtés à un problème de compréhension au tout début du projet. Nous avons pensé qu'il était nécessaire de pouvoir créer plusieurs tables dans ce programme. Nous avons donc commencé à élaborer un programme basé sur de multiples structures qui nous noyait dans sa complexité et que nous ne parvenions pas à faire fonctionner correctement. Cependant, après des discussions avec des camarades et notre professeur, nous avons compris notre erreur, ainsi nous avons pu avancer.

Ensuite l'une de nos difficultés a été l'utilisation des propriétés des types structurés avec des objets de type pointeurs. En effet, nous étions repris par Visual Studio qui produisait des erreurs que nous ne comprenions pas dans quelle situation il fallait utiliser l'opérande `.` et quand il fallait utiliser l'opérande `->`. Cependant, en assistant aux cours et TD d'IAP sur ce sujet, notre incompréhension a disparu, nous permettant de manipuler nos données avec bien plus de clarté.

Dans le sprint 5, nous nous sommes heurtés à un problème, en effet, notre lecture des Entrée-Sorties de ce sprint nous avait fait comprendre que la sélection des textes s'opérait uniquement sur la première lettre des mots. Notre sprint était 0-défauts pour le test mais une discussion avec un camarade nous a appris qu'il fallait faire autrement. Nous avons donc changé de méthode pour prendre en compte des mots complets dans la sélection. Finalement, notre solution finale était plus simple et plus performante que ce que nous avons codé dans un premier temps.

De même dans le sprint 5, lors de l'implémentation des dates, un souci de décalage est survenu. Après quelques essais et réflexion nous nous sommes rendus compte que le problème était causé par l'année, qu'il fallait absolument placer en dernier la composante de l'année car, étant plus longue que les autres composantes de deux caractères, elle provoquait un décalage très peu esthétique qui faussait notre fichier de sortie.



## Conclusion :

Ce projet collaboratif en programmation nous a permis de découvrir le langage C, ses possibilités, mais aussi ses aspects difficiles à appréhender. Ce projet, nous a également introduit au travail de codage et de développement en groupe, il nous a appris à suivre les directives de génie logiciel, à l'appliquer correctement et à tirer le maximum de notre environnement de développement pour maximiser l'optimisation de notre travail. Cela nous a aidé non seulement à maîtriser les outils de la programmation mais aussi à intégrer la rigueur nécessaire à la création de bons programmes. Tout au long de cette aventure, nous nous sommes sentis propulsés sur le terrain, le champ de bataille du programmeur qui remue ciel et terre pour satisfaire son client et honorer son contrat. Faire face aux imprévus, corriger des malentendus ... Ce fut très stimulant de se retrouver dans cette situation réaliste, le plaisir de la découverte succédant toujours à la fébrilité et au stress de la recherche.

Enfin, l'aspect collaboratif de ce projet fait partie de ses points forts. En effet, la motivation mutuelle est un moteur si puissant qu'il permet de surmonter tout type de désespoir. De plus, la puissance de deux esprits focalisés sur le même objectif, communiquant sur un problème est une des formes les plus efficaces de réflexion. Chacun comblant les faiblesses de l'autre, nous avons pu nous tirer mutuellement vers le haut. Le travail collaboratif est enrichissant car il permet d'apprendre autant de l'autre que de soi.



Annexe 1 : Jeu de données de test (inSp5.txt), sortie attendue (outSp5.txt), et sortie (run.txt) pour le sprint 5.

```
Create_table Pays 5 nom TEXT pib INT capitale TEXT latitude FLOAT longitude FLOAT Afficher_schema Pays
Insert_enregistrement Pays Turquie 10946 Ankara 39.92 32.86
Insert_enregistrement Pays Chine 6807 Pekin 39.92 116.44
Insert_enregistrement Pays Allemagne 45085 Berlin 52.52 13.41
Insert_enregistrement Pays Mexique 11180 Mexico 19.42 -99.12
Insert_enregistrement Pays Canada 51958 Ottawa 45.41 -75.69
Insert_enregistrement Pays Nigeria 3006 Lagos 6.45 3.39
Insert_enregistrement Pays Australie 67468 Melbourne -37.81 144.96
Insert_enregistrement Pays Bresil 11208 Brasilia -22.45 -49.23
Insert_enregistrement Pays Japon 38492 Tokyo 35.70 139.73
Insert_enregistrement Pays Zimbabwe 906 Harare -17.82 31.03
Insert_enregistrement Pays Russie 14555 Moscou 55.75 37.61
Insert_enregistrement Pays Maroc 3083 Rabat 33.97 -6.84
Insert_enregistrement Pays France 41421 Paris 48.85 2.35
Insert_enregistrement Pays Singapour 55182 Singapour 1.35 103.81
Afficher_enregistrements Pays
Delete_enregistrement Pays 10
Insert_enregistrement Pays Indonesie 3475 Jakarta -6.17 106.82
Delete_enregistrement Pays 1
Afficher_enregistrement Pays 10
Select_enregistrement Pays longitude -45 45
Delete_table Pays
Create_table Calendrier 3 evenement DATE description TEXT pays TEXT
Afficher_schema Calendrier
Insert_enregistrement Calendrier 14/10/1066 Bataille_Hastings Angleterre
Insert_enregistrement Calendrier 19/08/1274 Sacre_Edouard_I Angleterre
Insert_enregistrement Calendrier 07/10/1337 Guerre_Cent_Ans France
Insert_enregistrement Calendrier 07/06/1494 Traite_Tordesillas Espagne
Insert_enregistrement Calendrier 13/09/1515 Bataille_Marignan France
Insert_enregistrement Calendrier 10/08/1539 Ordonnance_Villers_Cotterets France
Insert_enregistrement Calendrier 13/04/1598 Edit_Nantes France
Insert_enregistrement Calendrier 02/09/1666 Incendie_Londres Angleterre
Insert_enregistrement Calendrier 21/01/1793 Mort_Louix_XVI France
Insert_enregistrement Calendrier 28/05/1936 Machine_Turing Angleterre
Select_enregistrement Calendrier pays D G
Select_enregistrement Calendrier evenement 07/10/1337 10/08/1539
Exit
```



Pays 5

nom TEXT

pib INT

capitale TEXT

latitude FLOAT

longitude FLOAT

1 TEXT Turquie INT 10946 TEXT Ankara FLOAT 39.92 FLOAT 32.86  
2 TEXT Chine INT 6807 TEXT Pekin FLOAT 39.92 FLOAT 116.44  
3 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41  
4 TEXT Mexique INT 11180 TEXT Mexico FLOAT 19.42 FLOAT -99.12  
5 TEXT Canada INT 51958 TEXT Ottawa FLOAT 45.41 FLOAT -75.69  
6 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39  
7 TEXT Australie INT 67468 TEXT Melbourne FLOAT -37.81 FLOAT 144.96  
8 TEXT Bresil INT 11208 TEXT Brasilia FLOAT -22.45 FLOAT -49.23  
9 TEXT Japon INT 38492 TEXT Tokyo FLOAT 35.70 FLOAT 139.73  
10 TEXT Zimbabwe INT 906 TEXT Harare FLOAT -17.82 FLOAT 31.03  
11 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61  
12 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84  
13 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35  
14 TEXT Singapour INT 55182 TEXT Singapour FLOAT 1.35 FLOAT 103.81  
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84  
2 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41  
5 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39  
9 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61  
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84  
11 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35

Calendrier 3

evenement DATE

description TEXT

pays TEXT

3 DATE 07/10/1337 TEXT Guerre\_Cent\_Ans TEXT France  
4 DATE 07/06/1494 TEXT Traite\_Tordesillas TEXT Espagne  
5 DATE 13/09/1515 TEXT Bataille\_Marignan TEXT France  
6 DATE 10/08/1539 TEXT Ordonnance\_Villers\_Cotterets TEXT France  
7 DATE 13/04/1598 TEXT Edit\_Nantes TEXT France  
9 DATE 21/01/1793 TEXT Mort\_Louix\_XVI TEXT France  
3 DATE 07/10/1337 TEXT Guerre\_Cent\_Ans TEXT France  
4 DATE 07/06/1494 TEXT Traite\_Tordesillas TEXT Espagne  
5 DATE 13/09/1515 TEXT Bataille\_Marignan TEXT France  
6 DATE 10/08/1539 TEXT Ordonnance\_Villers\_Cotterets TEXT France



Pays 5

nom TEXT

pib INT

capitale TEXT

latitude FLOAT

longitude FLOAT

1 TEXT Turquie INT 10946 TEXT Ankara FLOAT 39.92 FLOAT 32.86  
2 TEXT Chine INT 6807 TEXT Pekin FLOAT 39.92 FLOAT 116.44  
3 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41  
4 TEXT Mexique INT 11180 TEXT Mexico FLOAT 19.42 FLOAT -99.12  
5 TEXT Canada INT 51958 TEXT Ottawa FLOAT 45.41 FLOAT -75.69  
6 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39  
7 TEXT Australie INT 67468 TEXT Melbourne FLOAT -37.81 FLOAT 144.96  
8 TEXT Bresil INT 11208 TEXT Brasilia FLOAT -22.45 FLOAT -49.23  
9 TEXT Japon INT 38492 TEXT Tokyo FLOAT 35.70 FLOAT 139.73  
10 TEXT Zimbabwe INT 906 TEXT Harare FLOAT -17.82 FLOAT 31.03  
11 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61  
12 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84  
13 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35  
14 TEXT Singapour INT 55182 TEXT Singapour FLOAT 1.35 FLOAT 103.81  
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84  
2 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41  
5 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39  
9 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61  
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84  
11 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35

Calendrier 3

evenement DATE

description TEXT

pays TEXT

3 DATE 07/10/1337 TEXT Guerre\_Cent\_Ans TEXT France  
4 DATE 07/06/1494 TEXT Traite\_Tordesillas TEXT Espagne  
5 DATE 13/09/1515 TEXT Bataille\_Marignan TEXT France  
6 DATE 10/08/1539 TEXT Ordonnance\_Villers\_Cotterets TEXT France  
7 DATE 13/04/1598 TEXT Edit\_Nantes TEXT France  
9 DATE 21/01/1793 TEXT Mort\_Louix\_XVI TEXT France  
3 DATE 07/10/1337 TEXT Guerre\_Cent\_Ans TEXT France  
4 DATE 07/06/1494 TEXT Traite\_Tordesillas TEXT Espagne  
5 DATE 13/09/1515 TEXT Bataille\_Marignan TEXT France  
6 DATE 10/08/1539 TEXT Ordonnance\_Villers\_Cotterets TEXT France