

## : La compilation du projet :

le programme me donne tous les résultats de mon vote et cela sans souci.

Pour ce qui est des méthodes uninominale je me suis contenté de faire un tri de vote (la somme des colonnes ) en ne comptant les votes les plus fort pour les candidats et enfin de faire les comparatifs à la fin.

```
vboxuser@ubuntu:~/Downloads/REV_Party-20231216T204640Z-001/REV_Party/REV_Party/src/Csv$ gcc -o scrutin main.c lecture_csv.c matrice.c uninominales.c
vboxuser@ubuntu:~/Downloads/REV_Party-20231216T204640Z-001/REV_Party/REV_Party/src/Csv$ ./scrutin -i jugement.csv -m uni1
Somme colonne 5: 59
Somme colonne 6: 72
Somme colonne 7: 93
Somme colonne 8: 104
Somme colonne 9: 71
Somme colonne 10: 60
Somme colonne 11: 60
Somme colonne 12: 53
Somme colonne 13: 72
Somme colonne 14: 72
8
Mode de scrutin : uninominale à un tour, 10 candidats, 25 votants, vainqueur = Q01->4 - Veggie Burger, score = 14.53%
```

Même principe pour la méthode de l'uninominal à 2 tours, on prenait en compte le vote le plus fort pour les candidats, ensuite lorsqu'il nous restait les deux candidats on refaisait faire une lecture en prenant compte le vote le plus fort mais un cran en dessous.

Le programme effectue des calculs basés sur ce fichier de données [jugement.csv](#). Les résultats affichés indiquent les sommes des voix pour chaque colonne, suivies par l'annonce des vainqueurs pour différents types de scrutin.

Dans le mode de scrutin uninominal à un tour avec 10 candidats et 25 votants, le "Veggie Burger" est déclaré vainqueur avec un score de 14.53%. Ce mode de scrutin désigne un seul vainqueur basé sur qui a le plus grand nombre de votes au premier tour.

Pour le mode de scrutin uninominal à deux tours, le programme affiche des résultats pour plusieurs itérations. Au premier tour, avec 10 candidats, deux candidats se distinguent : "Fire cracker" et "Veggie Burger", tous deux avec un score de 14.53%. Au second tour, qui semble être une confrontation directe entre ces deux candidats, "Fire cracker" est déclaré vainqueur avec un score significativement plus élevé de 32%.

La différence entre les scores au premier et au deuxième tour pour le "Fire cracker" met en lumière la dynamique du scrutin uninominal à deux tours, où les préférences des électeurs pour les candidats éliminés au premier tour peuvent changer substantiellement les résultats au second tour.

```
vboxuser@ubuntu:~/Downloads/REV_Party-20231216T204640Z-001/REV_Party/REV_Party/src/Csv$ gcc -o scrutin main.c lecture_csv.c matrice.
c uninominales.c
vboxuser@ubuntu:~/Downloads/REV_Party-20231216T204640Z-001/REV_Party/REV_Party/src/Csv$ ./scrutin -i jugement.csv -m uni2
Somme colonne 5: 59
Somme colonne 6: 72
Somme colonne 7: 93
Somme colonne 8: 104
Somme colonne 9: 71
Somme colonne 10: 60
Somme colonne 11: 60
Somme colonne 12: 53
Somme colonne 13: 72
Somme colonne 14: 72
8
Mode de scrutin : uninominale à un tour, 10 candidats, 25 votants, vainqueur = Q01->4 - Veggie Burger, score = 14.53%
Mode de scrutin : uninominal à deux tours, tour 1, 10 candidats, 25 votants, vainqueur = Q01->5 - Fire cracker, score = 14.53%
Mode de scrutin : uninominal à deux tours, tour 1, 10 candidats, 25 votants, vainqueur = Q01->4 - Veggie Burger, score = 12.99%
Mode de scrutin : uninominal à deux tours, tour 2, 2 candidats, 25 votants, vainqueur = Q01->5 - Fire cracker, score = 32.00%
```

### condorcet Minimax :

un exemple d'utilisation d'une ligne de commande en utilisant les balises de main pour compiler et exécuter le programme de scrutin : condorcet minimax .

```
vboxuser@ubuntu: ~/Downloads/REV_Party-20231216T204640Z-001/REV_Party/REV_Party/src/Csv
vboxuser@ubuntu:~/Downloads/REV_Party-20231216T204640Z-001/REV_Party/REV_Party/src/Csv$ gcc -o scrutin main.c lecture_csv.c matrice.
c uninominales.c graph.c condorcet.c
vboxuser@ubuntu:~/Downloads/REV_Party-20231216T204640Z-001/REV_Party/REV_Party/src/Csv$ ./scrutin -i jugement.csv -m cm
Mode de scrutin : Condorcet, 10 candidats, 25 votants, vainqueur condorcet = Q01->8 - Cheese Burger
vboxuser@ubuntu:~/Downloads/REV_Party-20231216T204640Z-001/REV_Party/REV_Party/src/Csv$
```

### jugement majoritaire :

```
vboxuser@ubuntu:~/Downloads/REV_Party-20231216T204640Z-001/REV_Party/REV_Party/src$ ./jugementTest jugement.csv test.txt
Gagnant trouvé avec la méthode du jugement Majoritaire :
Mode de scrutin : Jugement majoritaire,vainqueur = Burger Black Pepper avec la mention Bien (28.00 %)
vboxuser@ubuntu:~/Downloads/REV_Party-20231216T204640Z-001/REV_Party/REV_Party/src$
```

### Les Tests :

On a consciencieusement mis en œuvre une série de tests unitaires pour chaque module de l'application de scrutin. Pour chaque méthode de vote - que ce soit le scrutin uninominal, le Condorcet, ou le jugement majoritaire - on a élaboré un `main` dédié qui nous permettait de valider l'intégrité et la fiabilité des résultats fournis par chaque algorithme. Cette approche rigoureuse nous a assuré que chaque composant fonctionnait indépendamment comme prévu, avant de les intégrer dans le système plus large. Grâce à ces tests, on a pu détecter et corriger des anomalies de manière proactive, garantissant ainsi que l'ensemble du programme fonctionnait avec précision et efficacité lors de son exécution finale.

## Un exemple de page HTML générée par Doxygen :

Cette page fait partie de la documentation du projet "Projet Avancé REV PARTY" et présente des détails sur par exemple le fichier `jugementMajoritaire.c`, qui implémente la méthode du Jugement Majoritaire pour des élections. La page inclut les dépendances du fichier, illustrées par un graphe de dépendance qui relie `jugementMajoritaire.c` aux headers et aux fichiers sources associés, notamment `jugementMajoritaire.h`, `lecture_csv.h`, et diverses bibliothèques standards en C comme `stdio.h`, `stdlib.h`, `string.h`, et `unistd.h`.

Cela montre que le fichier `jugementMajoritaire.c` est bien intégré dans l'ensemble du projet et dépend de plusieurs autres modules et bibliothèques pour fonctionner, reflétant une structure de projet complexe et bien organisée.

← → ↻ file:///home/vboxuser/Downloads/REV\_Party-20231216T204640Z-001/REV\_Party/REV Party/src/ ☆

# Projet Avancé REV PARTY

Main Page Classes ▾ Files ▾ 🔍 Search

## jugementMajoritaire.c File Reference

Functions | Variables

Implémentation de la méthode du Jugement Majoritaire pour des élections. [More...](#)

```
#include "jugementMajoritaire.h"
#include "lecture_csv.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
```

Include dependency graph for jugementMajoritaire.c:

```
graph TD
    JM["jugementMajoritaire.c"]
    JMH["jugementMajoritaire.h"]
    LCH["lecture_csv.h"]
    UN["unistd.h"]
    STDIO["stdio.h"]
    STDBOOL["stdlib.h"]
    STRING["string.h"]
    GLOBAL["global.h"]

    JM --> JMH
    JM --> LCH
    JM --> UN
    JM --> STDIO
    JM --> STDBOOL
    JM --> STRING
    JM --> GLOBAL
    JMH --> STDIO
    JMH --> STDBOOL
    JMH --> STRING
    LCH --> STDIO
    LCH --> STDBOOL
    LCH --> STRING
    GLOBAL --> STDIO
    GLOBAL --> STDBOOL
    GLOBAL --> STRING
```

Functions