

BEAUQUIER Quentin, MARIE Clément, MOMY Aurore, PORTELLI Angelo, ROSSI Valentin

SAE 1.01
Analyse de systèmes de vote



Compte rendu

26 novembre 2022

I. Introduction	2
II. Fichier d'entrée et de sortie	3
1. Qu'est-ce qu'un fichier d'entrée et de sortie	3
2. Constitution d'un fichier d'entrée	3
3. Fonction de récupération pseudo universel au projet	4
III. Scrutin Majoritaire à deux tours	5
1. Origine et Utilisation	5
2. Avantages et Inconvénients	5
3. Mise en œuvre	5
4. Notre solution au problème	6
IV. Vote par approbation	7
1. Origine et Utilisation	7
2. Avantages et Inconvénients	7
3. Mise en œuvre	7
4. Notre solution au problème	8
V. Vote par valeur	10
1. Origine et Utilisation	10
2. Avantages et Inconvénients	10
3. Mise en œuvre	10
4. Notre solution au problème	10
VI. Vote alternatif	12
1. Origine et utilisation	12
2. Avantage et Inconvénients	12
3. Mise en œuvre	13
4. Notre solution au problème	14
VII. Conclusion	16

I. Introduction

Cette SAE a pour but de reconstituer 4 systèmes de votes existant ou ayant existé parmi ceux du monde entier. L'objectif premier est donc de les recréer en C++. Les systèmes de votes que nous avons choisis sont tous unique à leur manière. Ils ont leurs propres avantages et inconvénients et sont utilisés dans des cas différents. Ils nous ont donc paru pertinents, et intéressants à mettre en œuvre, ce qui était selon nous un critère de choix.

Nous allons donc voir dans un premier temps, comment au travers de chaque programme on a pu transformer les fichiers entrés, en des bulletins correspondant à chacun de nos systèmes de votes. Puis par la suite nous verrons un à un chacun des systèmes choisis et comment on a décidé de les développer.

Nous commencerons par le **Scrutin Majoritaire à deux tours**, puis par le **Vote par approbation**. Nous poursuivrons avec le **Vote par valeur** puis nous terminerons par le **Vote alternatif** avant de conclure.

II. Fichier d'entrée et de sortie

1. Qu'est-ce qu'un fichier d'entrée et de sortie

Le fichier d'entrée permet de rediriger la saisie au clavier vers une lecture de fichier, les commandes *cin* vont directement chercher la saisie dans le fichier. De même pour le fichier de sortie, on redirige la sortie du terminal vers un fichier, les commandes *cout* vont directement dans le fichier de sortie et non le terminal.

2. Constitution d'un fichier d'entrée

Un fichier d'entrée se présente sous la forme suivante :

candidat 1
candidat 2
candidat 3
etc...

Nom
Prénom
vote
Nom
Prénom
vote
etc...

Les votes des votants dépendent du système de vote, on a par exemple :

ALI
Mdallah
2 1 4 3

ou

ALI
Mdallah
oui non non oui

...

Tout dépend des nécessités du programme, afin de mener à bien sa mission le type du vote peut donc être une *string*, un *unsigned*, un *vector<string>* ou un *vector<unsigned>* .

3. Fonction de récupération pseudo universel au projet

Nous avons donc mis au point deux procédures qui permettent de récupérer les candidats et les votes avec le nom et prénom des votants.

La première récupère les candidats et les met dans un vecteur jusqu'à tomber sur une ligne vide, celle-ci est la même dans tous les votes.

La seconde récupère le nom et le prénom des votants et les met dans un vecteur sous la forme 'Nom Prénom', ensuite elle récupère les votes de différentes manières en fonction des besoins du vote et du programme c'est donc cette partie qui change le plus en fonction des votes.

III. Scrutin Majoritaire à deux tours

1. Origine et Utilisation

Le scrutin uninominal majoritaire à deux tours est un vote simple sans pondération se déroulant sur deux tours au maximum. Le scrutin majoritaire fait partie des méthodes de désignation d'élus les plus anciennes. Il permet d'attribuer un ou plusieurs sièges à celui ou ceux qui ont obtenu le plus de voix.

Il est utilisé en France, au Costa Rica ou encore en Sierra Leone pour les élections présidentielles.

2. Avantages et Inconvénients

a. Avantages

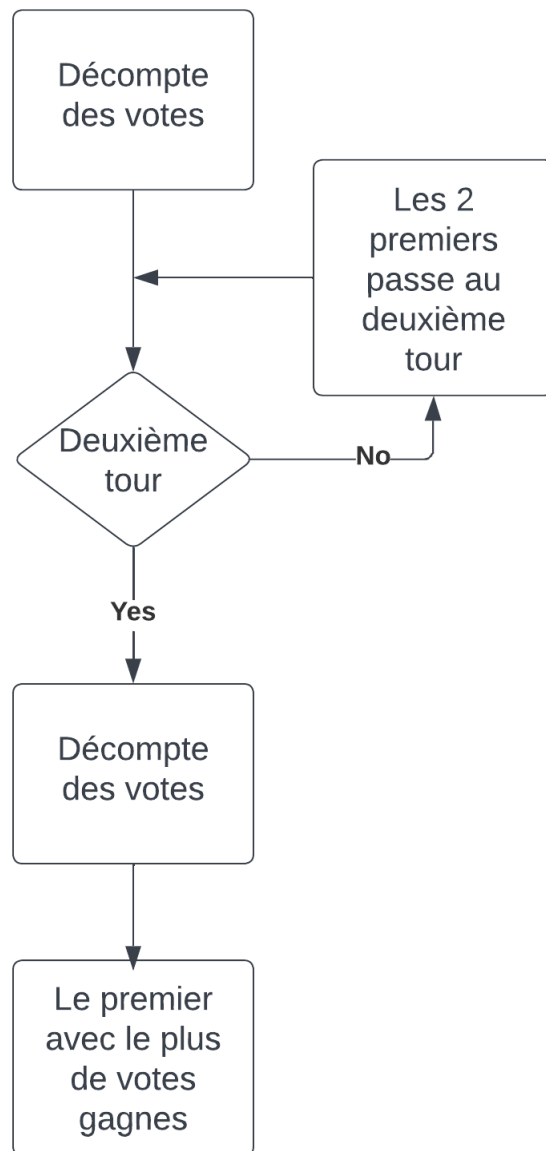
- Le scrutin majoritaire à deux tours est un vote facile à dépouiller.

b. Inconvénients

- impossibilité d'exprimer une opinion sur plusieurs candidats
- impossibilité d'exprimer un vote d'adhésion ou de rejet
- risque de tyrannie de la majorité

3. Mise en œuvre

Ce vote se fait en 2 tours, dans lesquels au bout du 1er tour il ne reste que 2 candidats, puis à la fin du 2ème tour il y a 1 vainqueur. Chaque tour seul les personnes avec le plus de votes gagne et les votes blanc sont comptés mais pas utilisés.



4. Notre solution au problème

Pour cela nous avons d'abord fait une fonction qui récupère dans un tableaux tous les candidats en rajoutant en première position le vote blanc, puis nous avons rajouté tous les autres candidats en lice dans ce tableau. Ensuite nous avons fait une fonction d'affichage qui affiche le candidat et son score. Nous avons créé une fonction pour récupérer les votes de chaque candidat. Pour finir nous avons fait la fonction de vote qui utilise la fonction d'affichage et la fonction de récupération de vote pour afficher les candidats et leurs scores. Puis nous trions la liste des candidats par rapport à leurs scores et nous affichons les votes et les qualifiés puis pour le 2ème tour nous vérifions si la taille du tableau de votes est égale à trois et nous affichons le premier car il a gagné.

IV. Vote par approbation

1. Origine et Utilisation

Le vote par approbation fut utilisé dans la République de Venise au XIII^{ème} siècle ainsi qu'au XIX^{ème} siècle en Angleterre. Il est aussi utilisé dans des associations notamment la Mathematical Association of America (MAA), l'association américaine de statistique (ASA), la société Social Choice and Welfare (SSCW) et dans certains pays d'Europe de l'Est ou de l'ancien bloc soviétique.

2. Avantages et Inconvénients

a. Avantages

- Très simple d'utilisation
- Elections rapides

b. Inconvénients

- Cas anti-démocratique (En cas d'égalité le premier dans la liste gagne)
- Imprécisions, ne reflète pas parfaitement la volonté des votant et doit être utilisé en petit comité

3. Mise en œuvre

Chaque votant vote parmi une liste de candidats, en précisant pour chacun s'il :

- Approuve
- Désapprouve

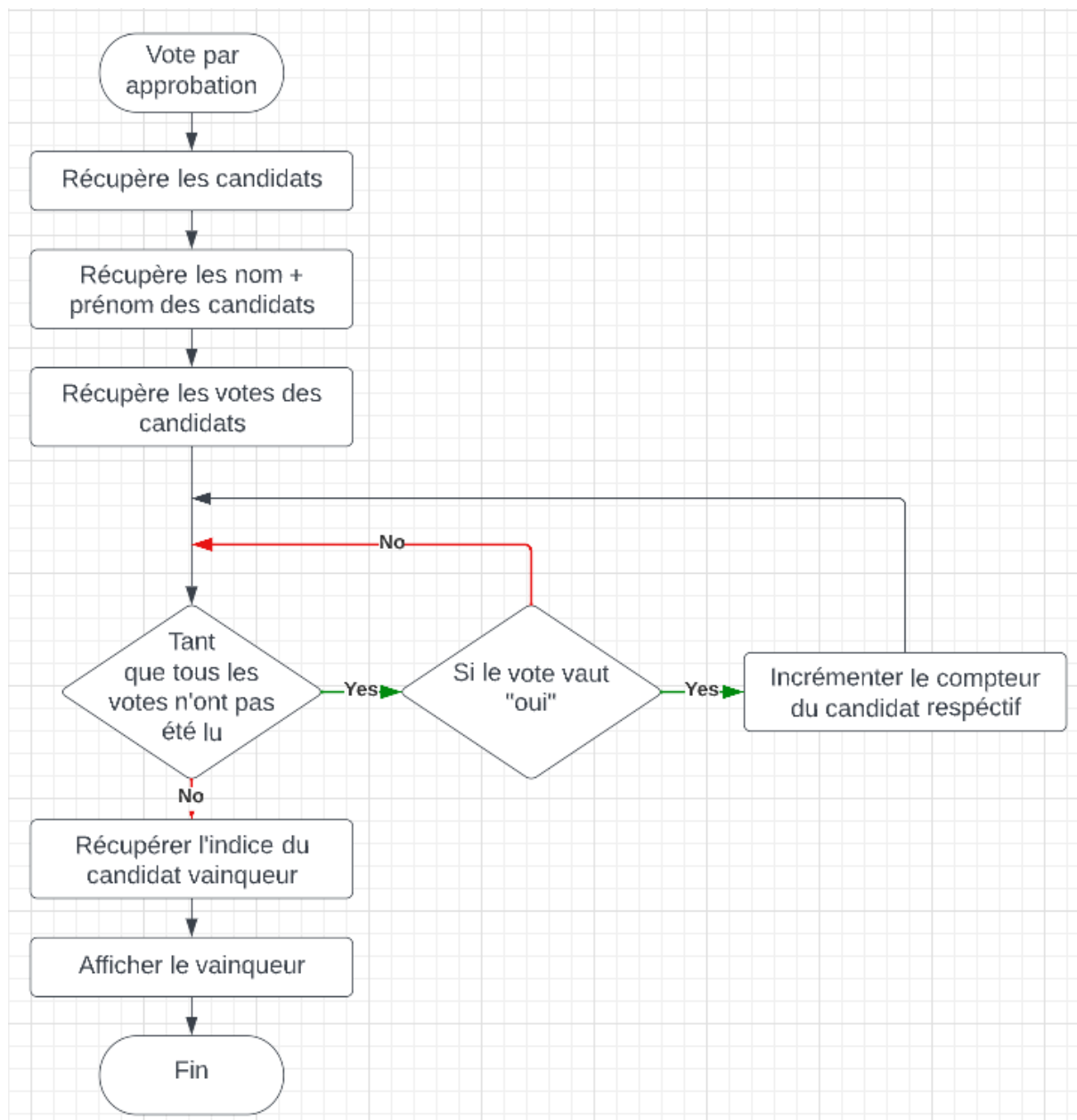
Il doit impérativement dire qu'il approuve ou non, les votes ou le votant n'approuve aucun candidat et les votes blanc ne sont donc pas comptés.

4. Notre solution au problème

3 fonctions sont utilisé pour faire fonctionner ce système de votes :

1. `recupVote` :
Expliqué plus haut dans le document, seule différence étant qu'elle récupère des string au lieu d'entier.
2. `voteApprobation` :
C'est la fonction unique pour le vote en question, tout d'abord elle récupère les candidats et vote à l'aide des fonction cité précédemment, ensuite elle incrémente un tableau de compteur parallèle à celui des candidat en fonction du vote lu, ceci est fait à l'aide de 2 boucles for car nous procédons avec une matrice, ensuite nous stockons l'indice du vainqueur grâce à la fonction `indiceDuPlusGrand` dans une variable, affichons dans le fichier de sortie les candidats et finalement affichons le vainqueur grâce à l'indice récupérer précédemment.
3. `indiceDuPlusGrand` :
Cette fonction sert simplement à trouver l'indice du plus grand élément d'un tableau, elle va nous être utile dans le programme principal de la fonction ci-dessous.

Algorithme résumé du code (le main) utilisant les 4 fonctions :



V. Vote par valeur

1. Origine et Utilisation

Le vote par valeur regroupe une famille de système de vote dont le principe consiste à associer à chaque candidat (ici jeux) une note. Cette dernière peut être sur 10 comme sur 100, etc., mais il est important de noter que plus la note est sur un nombre élevé plus le votant pourra nuancer son choix. Ce système de vote est notamment utilisé dans les jurys et les compétitions

2. Avantages et Inconvénients

a. Avantages

- Permet de nuancer les votes

b. Inconvénients

- Il a été conçu pour correspondre quasi exclusivement à l'environnement des jurys et des compétitions

3. Mise en œuvre

Chaque votant soumet une note sur 100 à chaque candidat avec 0 représentant un avis extrêmement défavorable envers le candidat en question et 100 un avis extrêmement favorable envers le candidat.

Le vote blanc n'est pas possible dans ce système de vote.

Après avoir recueilli tous les votes et les avoir placés dans un tableau de manière triée, on additionne toutes les notes données à chaque candidat pour ensuite les comparer. Le candidat ayant la note la plus élevée est donc élu.

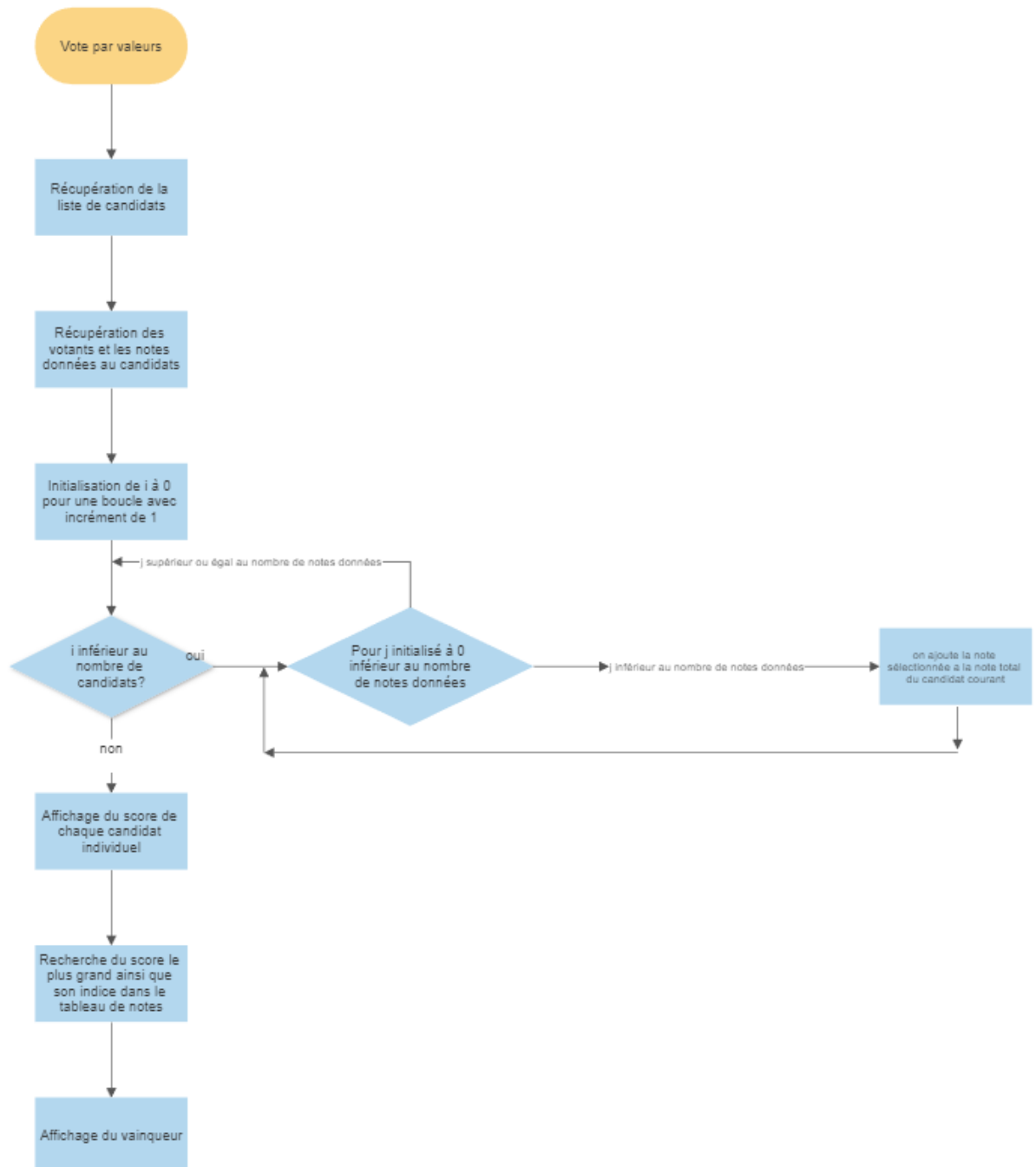
4. Notre solution au problème

La totalité des votes du fichier d'entrée sont récupérés par une fonction appelée "recupVote" et stockée dans une matrice d'entiers naturels avec chaque ligne correspondant à un votant et les notes qu'il a attribuées à chaque candidat dans le même ordre que le tableau contenant le nom des candidats. On récupère de même la liste des noms des candidats de l'élection en question.

La fonction effectuant l'analyse des résultats prend en argument la matrice et la liste mentionnée ci-dessus et crée un tableau nommé "notes" initialisé avec une taille égale au nombre de candidats présents. Son rôle est d'accueillir la note finale de chaque candidat. Pour récupérer ces notes, on crée deux boucles imbriquées pour parcourir la matrice de votes qu'on ajoute donc à l'emplacement du tableau "notes" associé au candidat sélectionné par les tours de boucle.

Une fois ce parcours fini, on injecte le nom de chaque candidat ainsi que leur score qu'ils ont obtenu à la fin de ce vote dans le fichier de sortie.

On recherche la note maximale du tableau de notes à l'aide de la fonction `max_element` provenant de la STL, puis, on trouve son indice à l'aide de la fonction "findIndex" pour trouver logiquement l'indice du nom du gagnant dans le tableau des candidats et l'injecter lui-même par la suite dans le fichier de sortie.



VI. Vote alternatif

1. Origine et utilisation

- a. Le vote à second tour instantané, aussi appelé vote alternatif, est un système électoral par classement. Il est notamment utilisé en Australie, à Nauru, en Papouasie-Nouvelle-Guinée pour l'élection des membres du parlement, ainsi qu'en Irlande, en Inde et au Sri-Lanka pour l'élection du président.
Le vote alternatif a été inventé en 1871 par l'architecte américain William Robert Ware, bien qu'en fait ce ne soit qu'un cas particulier du scrutin à vote unique transférable, lequel a été indépendamment développé dans les années 1850.

2. Avantage et Inconvénients

- a. Avantage
 - L'électeur peut s'exprimer sur l'ensemble des candidats et voter ainsi à la fois pour ses idées au travers d'un candidat, peut-être fortement minoritaire, *et* pour le candidat éligible qui lui convient le mieux ou lui déplaît le moins.
 - L'obligation d'obtenir une majorité absolue augmente la légitimité de l'élu.
- b. Inconvénients
 - Dans sa forme absolue, l'électeur se doit de classer tous les candidats, le dépouillement est alors difficilement faisable à la main, ce qui entraîne un handicap important à sa mise en œuvre.
 - Dans sa forme absolue l'acte de vote peut se transformer en pensum. En Australie où il est en vigueur, l'électeur préfère souvent obéir aux consignes de classement du parti de son choix plutôt que de réellement choisir un classement qui refléterait son opinion, ce qui contredit totalement le principe de ce vote.

3. Mise en œuvre

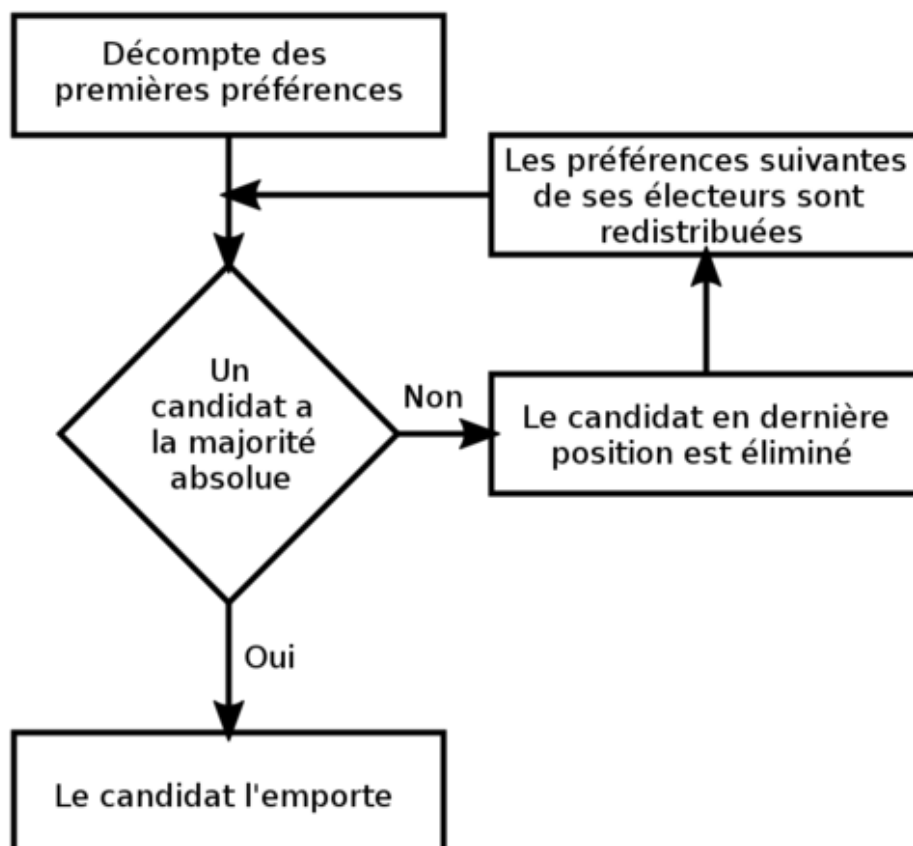
Le bulletin d'un vote alternatif est la liste de l'ensemble des candidats classés selon l'ordre de préférence de chaque votants.

A chaque tour on compte le nombre de premier choix par candidat, si l'un d'eux obtient la majorité absolue il est désigné vainqueur.

Dans le cas contraire, on regarde le candidat qui a reçu le moins de premier vote et on l'élimine. Les bulletins où le candidat éliminé était en premier choix sont redistribués en fonction du choix suivant du votant.

On passe ainsi au tour suivant en regardant si parmi les candidats restants quelqu'un à obtenue la majorité absolue et ainsi de suite.

Fonctionnement d'un vote à second tour instantané



4. Notre solution au problème

Pour pouvoir mettre en place ce système de vote nous avons divisé le problème en plusieurs fonctions.

a. `compterNbDeVote`

Tout d'abord nous avons écrit une fonction qui va compter le nombre de premier choix parmi notre liste de bulletins. Le score de chaque candidat est alors rangé dans une liste d'entier par rapport à une liste de référence contenant le nom de chacun des candidats (ici jeux).

b. `majoriteAbsolue`

renvoie un booléen vrai si le candidat entré en paramètre a obtenue la majorité absolue ce tour ci faux dans le cas contraire

c. `candidatAEliminer`

Cette fonction là, recherche l'élément minimum dans une liste d'entier et return l'élément de même indice de notre tableau référence. Comme ils sont tous deux rangé de façon à être en corrélation on obtient alors le nom de candidat ayant obtenu le moins de vote ce tour ci.

d. `indiceCandidat`

parcours un bulletin ou notre liste de référence afin de retourner l'indice du candidat rechercher

e. `supprimer`

Supprimer regroupe 2 fonctions: la première supprime dans un tableau un élément dont l'indice est entré en paramètre, tandis que la seconde parcourt un tableau de tableau pour appliquer la première fonction sur chacun de ces éléments. Ainsi on peut supprimer un candidat dans l'ensemble de nos bulletins.

f. `depouiller`

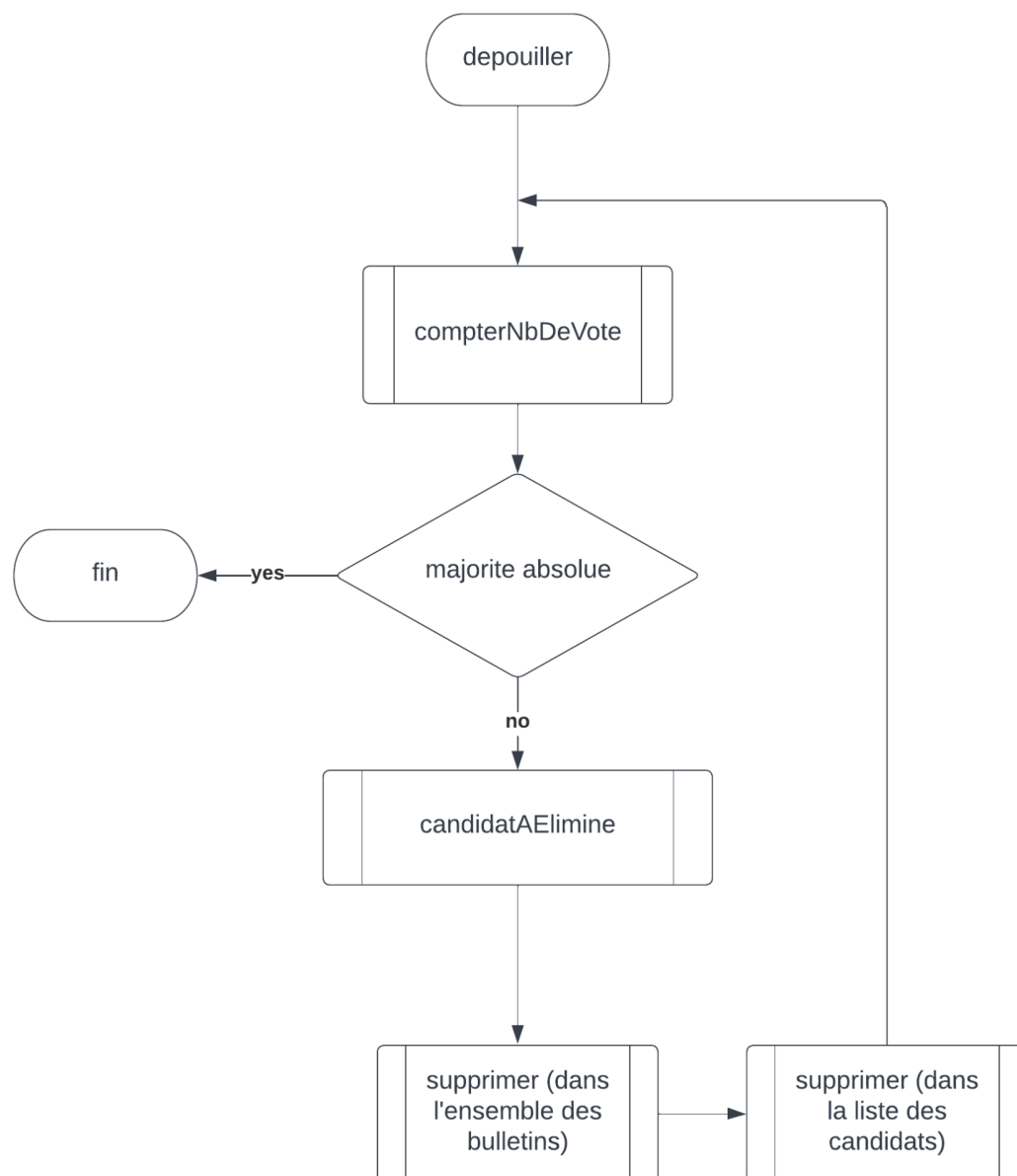
A chaque début de tour on commence par compter le nb de premier choix obtenu par chacun des candidats.

On entre ensuite dans une boucle qui va tester si chacun des candidats a obtenu la majorité absolue. Si c'est le cas le programme s'arrête et le candidat en question est désigné vainqueur, dans le cas contraire si personne n'a obtenu la majorité absolue on sort de la boucle et on continue le programme.

A ce stade, on regarde alors qui a obtenue le moins de premier choix. Le candidat concerné est alors éliminé, et on le supprime de la liste de candidat ainsi que sur chacun des bulletins. Sur ceux où il était premier, le second choix devient alors le nouveau premier choix.

Le tour est donc maintenant terminé on rappelle alors notre fonction depouiller avec cette fois ci en paramètre les nouvelle liste de candidat et de bulletin.

Dans cette version on ne redistribue à proprement parler les bulletins vu qu'on recompte à chaque tour.



VII. Conclusion

Cette SAE nous a permis de développer notre esprit d'équipe, notamment en partageant notre point de vue et les difficultés associées à chaque système de vote. En partageant nos ressources avec les autres, on a pu par conséquent progresser et approfondir notre maîtrise du langage ainsi que nos connaissances.

Le fait d'entreprendre un projet complet plutôt que de simples exercices demande une vision plus globale du problème. Grâce à l'esprit de chacun, on a su prendre plus de recul et trouver des solutions mieux structurées et plus optimisées pour nos programmes.