Jeu du Mindstermind

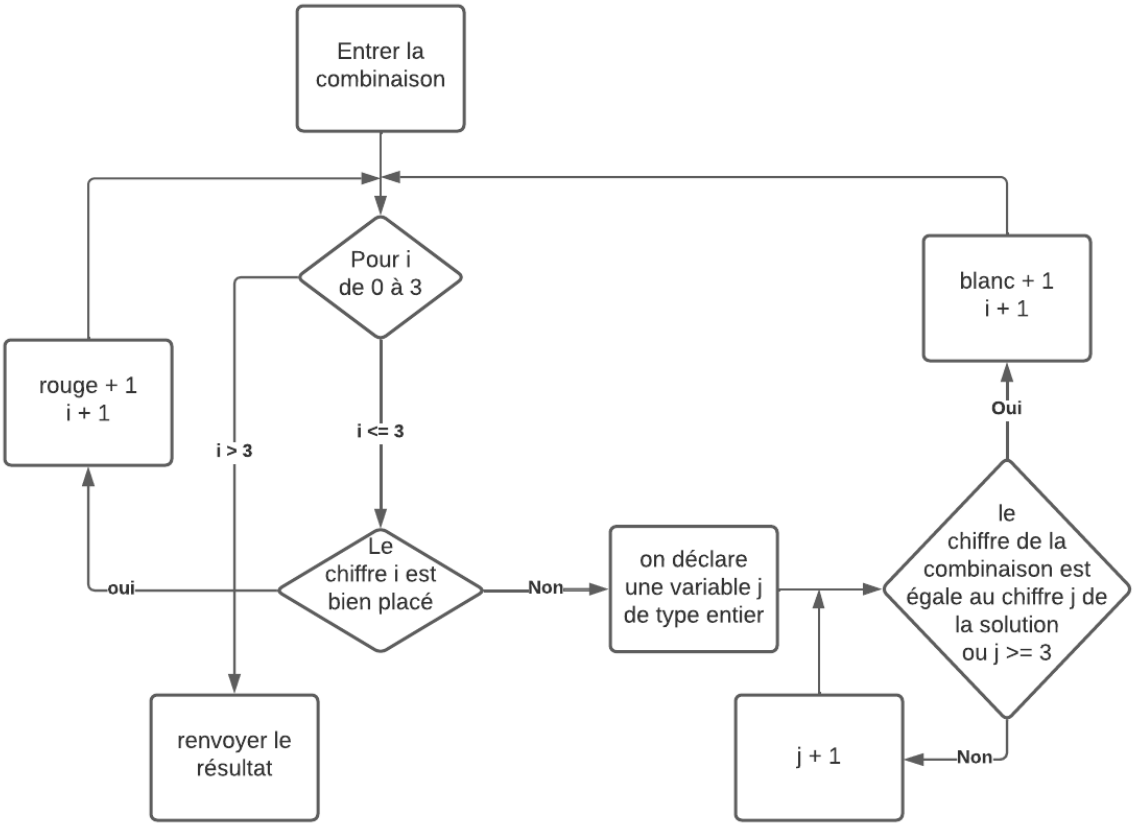
Le but de ce tp est de réaliser le jeu du mindstermind en langage C

Il te faudra compléter le fichier main.c, dans la fonction main du dois écrire une boucle qui vérifie que la fin du jeu : faire jouer le joueur tant que le jeu n'est pas fini. Dans la fonction check toi dois écrire un algorithme qui calcule le score du joueur, les points rouges comptent le nombre de chiffres bien placés et les points blancs comptent le nombre de chiffres mal placés.

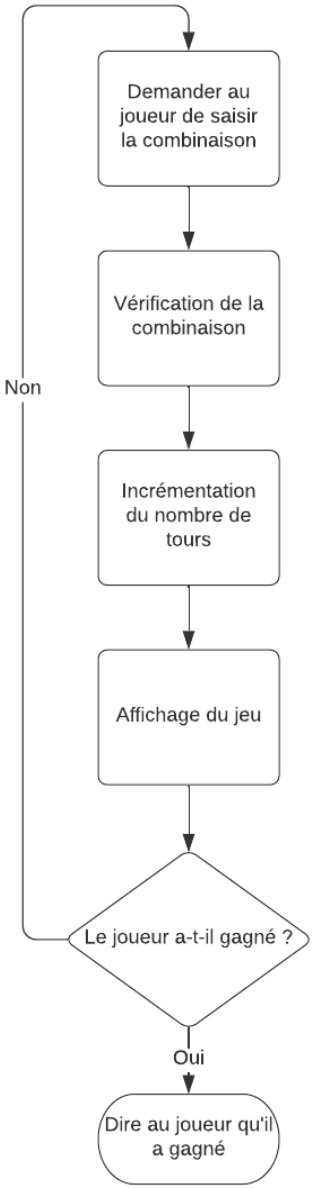
Par exemple, ici la solution est 2 0 3 4. On peut voir que le 2 est bien placé et que 3 0 sont mal placés. On a donc 1 point rouge et 2 points blanc.

```
                Result
|2|3|1|0|       | ••·|
```

L'algorithme de vérification :



Algorithme du jeu :



Pour demander au joueur de saisir on combinaison on utilisera la commande :

```
printf("\nEntrer les 4 valeurs\n");
scanf("%d %d %d %d", &send[try][0], &send[try][1], &send[try][2], &send[try][3]);
```

Pour vérifier la combinaison on utilisera la fonction check, en lui passant en paramètre les données saisis et attendus :

```
check(send[try], excepted);
```

Pour afficher le jeu, il faut utiliser la fonction display() :

```
display(send);
```

Pour dire au joueur qu'il a gagné, on utilise la fonction printf()

Memento du C

Pour rappel toutes les instruction (sauf la création de fonctions, boucles et conditionnelles) finissent par un ;

Les types :

- Int : nombre entier
- Char : caractère
- Double : nombre décimal
- Bool : booléen (true ou false)

Pour déclarer une variable, il suffit d'écrire :

type nom = valeur ;

Les fonctions utiles :

- Afficher un message : la fonction printf
  - Afficher du texte : printf("du texte") ;
  - Afficher des variables : printf("*type* = ", maVariable); avec :
    - %d pour les entiers
    - %c pour les caractères
    - %f pour les décimaux
    - On ne peut pas afficher de booléen, il faut écrire une fonction
- Demander à l'utilisateur de rentrer une données : scanf("*type*", &maVariable), le & veut dire que l'on veut récupérer l'adresse mémoire de la variable (l'endroit où elle est stockée dans la mémoire ram)
- Modulo : calcul du reste d'une division : variable1 % variable2

Les tableaux :

les tableaux sont des variables qui peuvent contenir plus valeurs à la suite les unes des autres.
Déclaration : *type* nom[taille];
Un tableau commence à l'indice 0 et sa taille n'est pas modifiable.

Par exemple le tableau solution :

0	1	2	3
2	0	3	4

Pour accéder à la case 0, on écrit : solution[0], on peut alors modifier sa valeur et y accéder.
Pour initialiser un tableau, le seul moyen est d'écrire une boucle, par exemple :

```
for (int i = 0; i < 4; i++) {
  solution[i] = random( );
}
```

On créer une variable i qui vaut 0, tant que i < 4 on exécute on change la valeur de la i par une nombre aléatoire. Une fois la boucle terminée, on incrémente i.

Les boucles :

- Tant que : while(condition) { contenu }
- Faire tant que : do { contenu } while (condition) ;

- Pour : for (int i = 0 ; i < nombreDeTours ; i++) { contenu }

Les conditionnelles :

- Si alors : If (condition) { contenu }
- Si alors sinon : If (condition) { contenu} else {contenu }

On peut aussi imbriquer les si alors sinon pour faire des si alors sinon si, sinon si, sinon (avec autant de cas que l'on veut)

Condition à rentrer :

- Egalité : variable == autre variable ou constante (2 =)
- Inférieur : variable < autre variable ou constante
- Inférieur ou égale : variable <= autre variable ou constante
- Supérieur : variable > autre variable ou constante
- Supérieur ou égale : variable >= autre variable ou constante
- Différent : variable != autre variable ou constante

On peut ensuite imbriquer les conditions pour vérifier plusieurs choses :

- Et : condition 1 && condition 2
- Ou : condition 1 || condition 2
- Non : ! conditon

```
do {
  if (rouge > 4) {
    gagne = true;
  } else {
    jouer();
  }
} while (!gagne);
```

Par exemple on peut écrire :
Ce code commence par vérifier si la valeur de rouge est supérieure à 4, si oui on change la valeur de gagne à vrai, sinon on appelle la fonction jouer. Tant que gagne n'a pas la valeur vraie, on recommence.

Raccourcies :

- score = score + 1 revient à écrire score++
- score = score - 1 revient à écrire score--
- score = score + 2 revient à écrire score += 2
- score = score * 2 revient à écrire score *= 2
- score = score / 5 revient à écrire score /= 5 ;
- score = score - 2 revient à écrire score -= 2