

	<p align="center">TP MASTERMIND</p>	<p align="center">BTS SN IR</p>
---	--	--

Présentation :

Le **Mastermind** est un jeu de société pour deux joueurs dont le but est de trouver un code. C'est un jeu de réflexion, et de déduction, inventé par Mordecai Meirowitz dans les années 1970 alors qu'il travaillait comme expert en télécommunications.

Il se présente généralement sous la forme d'un plateau perforé de 10 rangées de quatre trous pouvant accueillir des pions de couleurs. Le nombre de pions de couleurs différentes est de 8 et les huit couleurs sont généralement: rouge ; jaune ; vert ; bleu ; orange ; blanc ; violet ; fuchsia.



Principe

Un joueur A commence par placer son choix de pions, sans qu'il soit vu de l'autre joueur B, à l'arrière d'un cache qui les masquera à la vue de celui-ci (B) jusqu'à la fin de la manche. Le joueur B qui n'a pas sélectionné les pions doit trouver quels sont les quatre pions, c'est-à-dire leurs couleurs et positions.

Pour cela, à chaque tour, le joueur B doit se servir de pions pour remplir une rangée selon l'idée qu'il se fait des pions dissimulés. Une fois les pions placés, l'autre joueur A indique :

- Avec des marqueurs rouges, les pions de la bonne couleur bien placés en utilisant le même placement que les pions recherchés.
- Avec des marqueurs blancs, les pions de la bonne couleur, mais mal placés, avec les pions blancs.
- Ces marqueurs seront positionnés, sur une ligne contigüe, aux mêmes endroits que les pions identifiés.

La tactique du joueur B consiste à sélectionner en fonction des coups précédents, couleurs et positions, de manière à obtenir le maximum d'informations de la réponse du joueur A puisque le nombre de propositions est limité par le nombre de rangées de trous du jeu.

Adaptation de la problématique :

Pour l'élaboration du programme de jeu, les couleurs seront remplacées par des chiffres de **0** à **9**, la combinaison secrète sera composée de **6** chiffres, et les marqueurs seront des '**v**' ou '**x**' :

- '**v**' : pions de la bonne couleur bien placés.
- '**x**' : pions de la bonne couleur, mais mal placés.

La combinaison secrète sera proposée par le microprocesseur.

Pour réaliser ce programme de jeu, on se propose de mettre en œuvre une classe **CMastermind**.

Un .exe de l'application afin de réaliser des tests préalables ?

La Classe CMastermind :

➤ **Attributs :**

- *m_secret* : Code secret généré par l'ordinateur.
- *m_essai* : Code saisi par le joueur
- *m_code* : Code relatifs aux marqueurs ('v' ou 'x'). '_' non présent.
- *m_nb_essais* : Compte les essais successifs
- *m_nb_essaisMax* : Nombre d'essais max par partie
- *m_fini* : Indicateur partie finie
- *m_taille_code* : Taille du code à rechercher (max 6 chiffres)

➤ **Méthodes :**

- **usage()** : Affiche un texte informatif sur les règles du jeu.
- **choisirSolution()** : Détermine aléatoirement la combinaison secrète et initialise une nouvelle manche. Affecte *m_secret*
- **saisirEssai()** : Récupère le code saisi par le joueur, affecte *m_essai* et *m_nb_essais*.
- **verifierEssai()** : Vérifie si la combinaison proposée par le joueur correspond à la combinaison secrète et détermine le placement des marqueurs. Affecte *m_code*.
- **afficherResultat()** : Affiche si la partie est finie/pas finie, le nombre d'essais restants, le code relatifs aux marqueurs *m_code* et le code secret *m_secret* lorsque la partie est terminée (gagnée ou perdue).
- Le constructeur par défaut affecte automatiquement une taille de code de 6 et donne 8 essais.
- Le constructeur surdéfini doit permettre de modifier la taille du code et le nombre d'essais.

CMastermind
<i>attributs</i>
-m_secret : String
-m_essai : String
-m_code : String
-m_nb_essais : int
-m_nb_essaisMax : int
-m_fini : boolean
-m_taille_code : int
<i>opérations</i>
+CMastermind()
+CMastermind(: int, : int)
+usage() : void
+choisirSolution() : void
+saisirEssai(: String) : void
+verifierEssai() : void
+afficherResultat() : void
+get_m_fini() : boolean

Algorithmes, organigrammes et extraits de code source disponibles :

- Code source relatif à la méthode **choisirSolution()**

```
srand(time(NULL));  
  
m_secret = to_string(rand() % 999999 + 100000);  
m_secret.resize(m_taille_code);
```

Directive `#include <time.h>` à intégrer.

- Algorithme relatif au programme principal :

DEBUT

Instancier un objet « Partie »
Choisir une solution

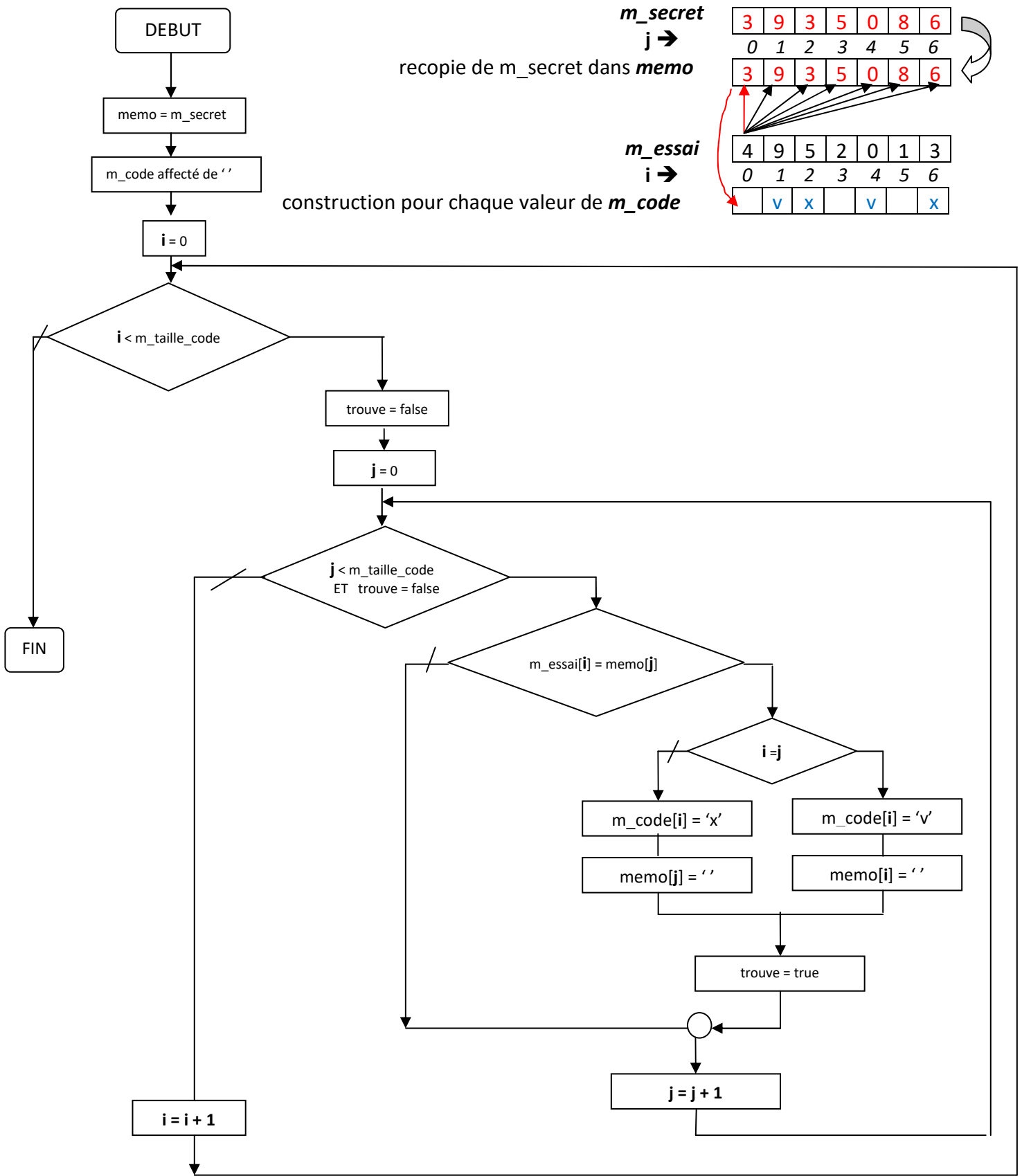
FAIRE

Saisir votre proposition de code
Vérifier votre proposition
Afficher les résultats

TANT QUE « La partie n'est pas finie »

FIN

- Organigramme relatif à la méthode **verifierEssai()**:



- Algorithme relatif à la méthode ***afficherResultat()***:

DEBUT

SI (code marqueurs = « vvvvvv »)

ALORS Afficher « Vous avez gagné »
 Afficher « Le code secret était »
 Afficher le code secret
 Partie finie = vrai

SINON SI (nbre essais = nbre essais max) ET (partie pas finie)

ALORS Partie finie = vrai
 nbre essais = 0
 Afficher « Manche finie – Vous avez perdu »
 Afficher « Le code secret était »
 Afficher le code secret

SINON

 Afficher « Incorrect – Il vous reste »
 Afficher nbre essais restants
 Afficher le code marqueurs
 Initialiser le code marqueurs avec des espaces

FIN

Questions :

- Créer la classe CMastermind.
- En consultant la classe string (<http://www.cplusplus.com/reference/string/string/>), comprendre le code relatif à la méthode ***choisirSolution()***.
- En utilisant au mieux les méthodes de la classe ***string***, coder toutes les méthodes de la classe.
- Coder le programme principal.
- Créer une nouvelle partie qui puisse se jouer avec un code de 4 chiffres en 6 essais.