        - construire une architecture de fichiers - construire un programme
(fichiers .h et .c; stdin; stdout, stderr)

        - construire un fichier: include, main, printf
```
#include <stdio.h>
int main()
{
        printf("HelloWorld\n");
}
```
        - architecture de fichiers: Define.h pour la définition des constantes
```
#define NB_NEURONS 10
\\
#include "Define.h"
printf("Nb Neurons: %d\n",NB_NEURONS);
```

        - les macros:
```
#define printN(number) printf("number : %d\n,number);
\\
printN(NB_NEURONS);
```

- intéret du préprocesseur:
(permet de passer d'un fichier avec des predirectives et du code C, à du code C seul)

- à la fin du cours: donner les polys sur le réseau de Hopfield

2eme TD: Init du réseau de Hopfield

        - Rappel:
-> inspiration de la physique (théorie des spins de glass en magnétisme)
-> modélisation des poids des synapses par des poids réels entre -1.0 et 1.0
-> modélisation de l'état des neurones par une variable bipolaire (1.0 : actif, -1.0 : ina
ctif)
        - Ex:
-> modélisations des structures associés(vecteurs double* et matrices double**)
-> structure des pointeurs = tableau d'objets
-> définition de l'opérateur de multiplication entre un vecteur et une matrice
-> initialisation des structures (opérateur malloc, + mise des valeurs à 0.0)
-> test du fonctionnement de l'opérateur* sur les structures initialisées

3eme TD: Apprentissage et fonctionnement du réseau

        - Rappel:
-> Loi de Hebb (Toute activité entre 2 neurones qui implique une action causale voit la co
nnection entre les neurones renforcés)
-> Traduction au niveau du réseau (correlation des patterns, projection de Hertz, Krogh &
Palmer, 1991)
-> fonctionnement du neurone à seuil

        - Ex:
-> Mise en place des patterns, en dur (10 neurones, 1pattern)
-> Calcul des poids, à partir des patterns (NB_PATTERNS)
-> Algorithme de fonctionnement, critère d'arret MAX

4eme TD: Lecture sur un fichier

        -Rappel:
-> Notion de descripteur de fichier (FILE*)
-> ouverture d'un fichier (fopen)
-> lecture des données d'un fichier (fscanf, ou fgetchar, \n, EOF)
-> fermeture du fichier (fclose)

        -Ex:
-> mise en place des fonctions dans File.c des procédures permettant la lecture du fichier
 TabProtos0.txt)
-> modification du nombre de neurones, et du nombre de patterns
-> construction d'un objet intermédiare (double** TabPatterns), à partir duquel on constru
it les poids par corrélation hebbienne
-> vérification du fonctionnement

5eme TD: Ecriture sur un fichier

-Ex:
-> mise en place d'un fichier dans lequel on ecrit l'état initiale du réseau(avec des # p
our 1, ' ' pour 0)
-> ecriture du pattern obtenu apres le calcul dans le même fichier, à la suite
-> écriture de l'état du réseau à chaque reverbération intermédiaire (apres chaque mise à
jour)

6eme TD: Bassins d'attraction et bruit

-Rappel:
-> bassin d'attraction: cuvette d'énergie par analogie à la physique (notion d'équilibre s
table/instable)
-> influence du bruit sur un réseau attracteur: permet de retrouver le pattern le plus pro
che stocké

-Ex:
-> génération du bruit dans le modèle:par inversion d'un pattern stocké selon un certain p
ourcentage
-> méthode srand et rand (stdlib.h/random.h), time (time.h)
-> analyse des patterns intermédiaires
-> tester le nombre de patterns stockés

7eme TD: Modèle à graded-response (Hopfield,84)

-Rappel:
-> Intérêt de neurone sigmoidal: sortie = fréquence d'émission
-> Modèle de Hopfield avec graded-response

-Ex:
-> mise en place du modèle, soit avec une fonction sigmoidal, soit avec une fonction linéa
ire
-> influence sur les réponses (rajouter dans le fichier sortie un autre symbole (°) si -1.
0< & <1.0)