# Cours 1

En C, il est primordiale de déclarer une méthode avant de l'utiliser.

Macro/Symbole: La valeur est remplacer par sa valeur avant la compilation. Substitution.

Variable: Prend de l'espace mémoire.

# Cours 2

TP : Notation Hongroise + Alignement

P.20 Les principes du TP

# Cours 4

FILE \*fin = NULL;

FILE \* fout = stdout;

...

fin = fopen("./file.in","+");

...

fclose(fin) (il faut d'abord verifier si fin == stdin, car on ne peut fermer ce dernier)

Tee c'est un branchement qui rentrer 1 fois et sort 2 fois. Branchement en Y.

Makefile:

<cible> c'est le fichier que tu veux creer (executable)

On met le .c dans la première dépendance pour qu'il sache si le fichier a été modifier.

Autre facon de faire :

tp1 : tp1.c

-> gcc -g -o tp1 tp1.c (on peut rajouter -Wall, -std=c11, -Wextra, -pendantie)

.PHONY: tu met apres tout les dependance qui ne prosuise pas de fichier sur le disque

--------

int tab[10[ = {1, 2, 3 ...}

tab++

printf("%d", tab[i])

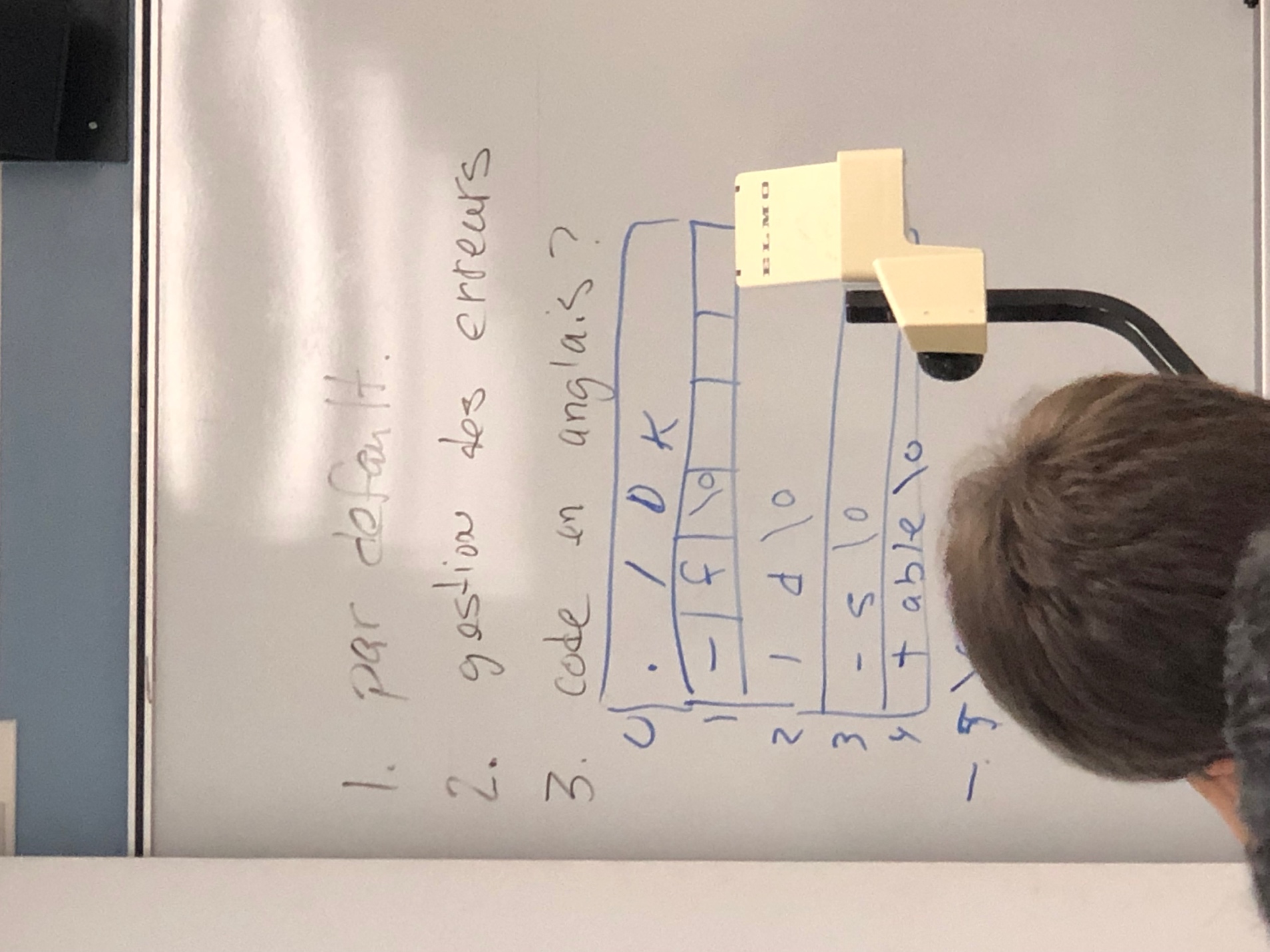
tab = tab + 7; //Deplace le pointeur.

-------

p.28

adresse de a[2] – adress de a = 2. Car l'adresse de a est celle de la premiere case du tableau qui est 0. Ca peut etre 10 et 12 ou peu importe. T'es simplement a 2 de plus loin.

------



-------

p.36

Pour TP – choisir de couper les noms trop long ?

-----

Le sizeof d'un ptr vas toujours retourner 64b

-----

TEXT : code en memoire

# Cours 5

Même si un variable *matrice* est un pointeur, le fait de faire *matrice[ ][ ]* est déjà en soit un référencement.

---

char \*mois[] = {"bla","blabla"}

*mois* est une matrice qui prend prendre des string de taille x.

---

p.12, le i représente la ligne dans la matrice.

---

malloc et les doubles [][] permettent d'allouer de l'espace en mémoire. Le \* ne reserve qu'un espace.

---

Pour les structure :

p.x quand c'est une copie

p->x quand c'est par reference

---

typedef enum sexe{M=1, F=2} sexe\_t;

Ici, sexe\_t est le nouveau type cree.

size\_of the sexe\_t prend les entier p.27. Voir ce que fait size\_of un type

---

malloc retourne un adresse

# Cours 5

Dans typedef Union Point\_s :

char Nom[50]

char Ville[30]

char Province[40]

La taille de l'Union est de 50. L'Union écrase les valeur stocker après leur utilisation pour ne travailler que sur la mémoire la plus grande. Le but est de minimiser l'espace requis.

Struct : n'a pas de fonction. Sinon, c'est une Class (c++).

Pile : First in, last out

Dans créer Pile, il se peut que l'on fasse de l'allocation dynamique. Il faudra Détruire l'espace alloue.

Tu peux faire :

struct StackNode {

} Stack;

Puis tu fais : Stack root = stackCreate();

Et tu peux faire :

Struct StackNode root = stackCreate();

Si tu ne fais pas de :

typedef struct {

} Stack;

Ou bien comme dans le premier "Tu peux faire".

File : First in, first out

A.first = NULL

A.last = NULL

Avantage/Inconvenients si on choisit une liste dynamique ou un tableau dynamique ?

Avantage : insertion

Inconvenient : recherche (et donc supression)

# Cours 6 – Avant intra

#IF FRANCAIS

... fr.dic

#ENDIF

#IF ANGLAIS

... ang.dic

#ENDIF

puis sur la ligne de commande : $ gcc -DFRANCAIS fichier.c

-----

Il n'y a pas de surcharge en C, alors il faut donner un autre nom.

----

# Cours 7

Dans $((i + i)), le $ prend le contenue de la variable de retour et le ramène.

Marche comme $(($a + $b)).

Ya pas de float en bash, on compte par 1.

Tu peux faire :

tab[0] = pomme

tab[1] = poire

tab[2] = fraise

et plus tard, tu peux même rajouter

tab[4] = pêche

Tableau dynamique sans pré initialisation

$? pour retourner l'état d'un programme (code de retour)

ls : echo $? affiche le contenu du dossier suivit de 0 (code pour que ça marche bien)

En Octale, tout ce qui est au-dessus de 07 n'existe pas.

Dans l'exemple de Crontab, le \*/10 \* \* \* implique qu'il est utilisé à chaque 10 minutes.

gcc -1<chemin> pour intégrer une librairie (puis tu l'include dans ton .c)

P.17 Importante -L Construction d'une librairie dynamique

Si tu veux éviter les dépendances system = .a

Tu sais pas c'est quoi l'environnement

Pour les static

Edition de lien (Compilation)

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Si tu veux éviter le volume = .o

Environnement local

Exécution

Dynamique

Les Assert arrêtent le programme – ne les utiliser que dans les testes