Programmation C

Chap.1 : Introduction & Généralités

Bibliothéques utilisés :

- stdio.h
- Conio.h
- Math.h
- Stdlib.h

Printf("Chaine %d(entier) %p(adresse) %c(Char) %f(float) "
,variables(x,y….));
Scanf("%d%d%f", Adresses(&x,&y,&z));

&& => And
|| => Or
! => Not

Conversion : (float) x

X = getchar()
Putchar('caractére') ou putchar(code ascii(entier))
Et

Chap.2 : Les structures alternatives & répétitives

Max = a>b ? A : b ;

if (condition)
{

```
Traitment}
else
{
Traitement;
}

Switch (x)
{
Case valeur 1 : traitement;break;
Case valeur 2 : traitement;break;
}

While… ; Do while ..

For(init,test,avancement)
{
Traitement;
}
```

Continue : passer a la prochaine itération de la boucle
Goto : se deplacer vers un emplacement dans le programme
    Goto etiquette
    Etiquette : traitement

Chap.3 : Les Tableaux

Tableaux unidimensionels :
    Déclaration :
        int tab[50];

Float tab[50];
// déclaration prédefinie : int tab[5] = {1,2,3,6,8};
// int tab[] = {1,2,3};

Accées :

Tab[indice];

Remarques :

Tab retoutne l'adresse du premier élément

Tableaux multidimensionnels :

Déclaration :

Int mat[nlignes][ncolones];
//déclaration prédefinie : int mat[][] = {{2,2,3},{2,5,6}};

Accées :

Tab[ind_lignes][ind_col];

Chap.4 : Pointeurs

Pointeur sur variable :

<Type> *<NomPointeur> // déclare un pointeur <NomPointeur> qui peut recevoir des adresses de variables du type <Type>

Exemple:

int *p,v; //déclaration
P= &v; // affectation de d'adresse de 'v' a 'p'

Pointeur sur pointeur :

Exemple :

Int *p1,**p2,***p3,x;
P1 =&x;

P2=&p1;
P3=&p2;

Pointeurs et tableaux :

&tab[i] <=> (tab+i)

tab[i] <=> *(tab+i)

Tableaux a deux dimensions :

Int mat[nl][nc];

(int*) mat+i*nc+j <=> &mat[i][j]

((int) mat+i*nc+j) <=> mat[i][j]

Allocation dynamique de mémoire :

Tableau a une seule dimension :

```
Int n,*ptab ;
// Saisie de N
//Allocation de mémoire
Ptab = (int*) malloc(n*sizeof(int));
If ( !ptab //<=>ptab==NULL// ) exit(-1);
//Libération de la mémoire
Free(ptab);
//Réallocation de la mémoire
Ptab = (int*) realloc(tab,nouveau taille*sizeof(int));
```

Tableau multidimensionnel : (A revoir)

```
Exemple (3 dim : lignes, colone,epaisseur)
Int l,c,e,il,ic,ie,***ptab;
//saisie de nombre de lignes,col, et epaisseur
//allocation de memoire des lignes
Ptab = (int***)malloc(l*sizeof(int))
```

```
If (!ptab) exit(-1);
for(il=0; i<l; i++)
{
    *(ptab+i)= (int**) malloc (c*sizeof(int*));
    if(! *(ptab+i)) exit(-2);
    for(j=0; j<c;j++)
    {
        *(*(ptab+i)+j) = (int*) malloc (k*sizeof(int));
        if (! *(*(ptab+i)+j) ) exit(-3);
    }
}
//remplissage
for(i=0; i<l;i++)
    for(j=0; j<c; j++)
        for(m=0; m<k; m++)
            scanf("%d", *(*(ptab+i)+j)+m );
```

Chap.5 : Enregistrements & Structures

Définition d'une structure == Création d'un fichier '.h' :

```
Struct Nom_structure {
    Type champ1;
    Type champ2;
    …
}
Typedef stuct nom_structure ABREVIATION
```

Appel dans le programme principal :

```
#include "nomfichier.h"
```

```
Void main(){
    Struct Nom_structure nom_variable;
}
```

Accées de Lecture/ Ecriture a un champ :

```
Nom_var.nom_champs
(*adresse_var).nom_champs
Adresse_var->nomchamps
```