N'hésitez pas à demander à faire compléter ces notes (suggérez !)… ni à programmer !

Structure d'un programme C

- #include : inclusion de bibliothèques
- #define : définition de constantes
- typedef : définition de types.
- Variables globales : connues de toutes les fonctions…
- Prototypes : déclarent les fonctions.
- Fonctions : là où est le code
- Fonction main : là où est le code

Pas de variables globales en L2 pour apprendre à utiliser correctement les fonctions

Les prototypes sont obligatoires

La fonction principale, main est normalisée :

```
int main ( void )
{     variables de main… ;

      séquence d'instructions ;

      return 0 ;
}
```

Exemple de programme…

```
#include<stdio.h>

#define dim 10

typedef int vect[ dim ] ;

/* prototypes */
int somme( int n , vect data );

int somme( int n , vect data )
{     int k ;
      int S ;

      for( k=0, S = 0 ; k<n ; k++)
      {     S += data[k] ;
      }
      return S ;
}

int main( void )
{     int n = 3 ;

      vect x = {0,1,2,3} ;
      printf("S = %d\n", somme(n,x) );
      return 0;
}
```

Inclus la bibliothèque stdio (fonction d'entrée/sortie : ici printf). On utilisera aussi stdlib.h ou math.h. Noter les < > obligatoires.

défini le symbole dim comme valant 10

défini le type vect : un tableau de 10 entiers allant de 0 à 9

Indique au compilateur la présence d'une fonction appelée somme, retournant un entier et ayant deux paramètres : un entier appelé n et un tableau de type vect appelé data.

Boucle de sommation. Noter les 2 initialisations : celle du compteur k et de l'accumulateur S. Noter l'opérateur combiné +=

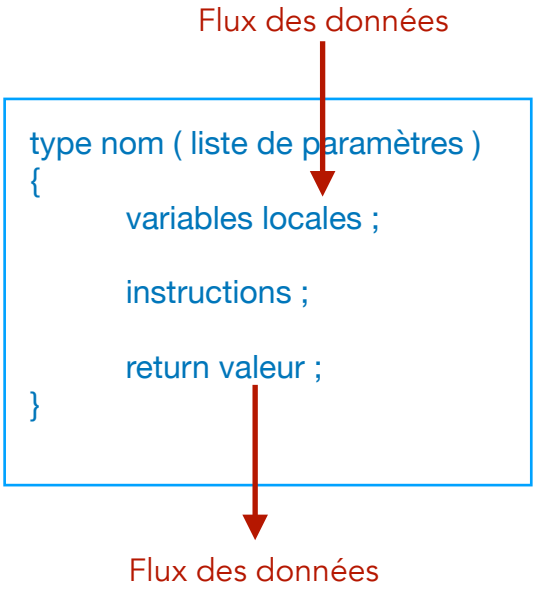
Fonction principale

Noter l'appel à la fonction somme dans le printf.

Formaliste de la fonction main obligatoire (en rouge)
Présence de la fonction main obligatoire

Pour printf, scanf, for, while, do…while, if et switch, se reporter au site pcastelan
Pensez à demander des améliorations le cas échéant.

FONCTIONS…



Les fonctions permettent de nommer une séquence d'instruction (définie par les { })

Les séquences d'instruction sont étanches, n'y existent que les variables qui sont créées entre les accolades : ce sont les variables locales.

Les fonctions permettent de définir des paramètres **qui sont des variables locales** auxquelles il est possible d'affecter des **valeurs** lors de l'appel de la fonction

Les fonctions permettent de retourner une valeur du même type que celui de la fonction.

Il est possible de créer des fonctions de type void (qui ne retournent rien (=> pas de return !) ou qui n'ont pas de paramètres (exemple la fonction main...)

Lorsque la fonction a fini d'être exécutée, toutes ses variables locales

VARIABLES…

- Une variable est le nom que l'on donne à une zone de la mémoire
- Une variable est typée :
  - Entier : int
  - Réel : float, double
  - Caractère : char

Utilisation	nom	&nom
Résultat	valeur contenue	adresse variable

- On accède au contenu de la variable à l'aide de son nom
- Une variable a un nom, un type, une valeur, une adresse…
- On créée une nouvelle variable en la déclarant comme suit : **type nom ;**
- **NB : on peut fixer la valeur de départ d'une variable : type nom = valeur de départ ;**

OPERATEURS…

- Arithmétiques : + ; - ; * ; / ; % (modulo)
- Comparaison : > ; < ; >= ; <=
- Unaires :
  - Incréments : ++ ; --
    - A++ ou ++A : augmente A de 1
  - Combinés : += ; -= ; *= ; /=
    - A += 3 ; // a = a + 3 ;
  - Négation : !A
    - En C est Vrai tout ce qui est non nul.

Il faut 2 & pour l'opérateur ET
Il faut 2 | (pipe) pour l'opérateur OU

A	B	A&&B	A||B
0	0	0	0
0	1	0	1
1	0	0	1
1	-1	1	1

SEPARATEURS…

; : sépare les instructions : A = 3 ; B = 4 ; …
, : sépare les éléments d'une liste
' ' : indique un caractère : char A = 'a' ;
" " : chaine de caractère : printf("coucou") ;
{} : limites d'une séquence.
. : partie décimale d'un réel : float a = 3.14 ;
[ ] : sélection d'un élément de tableau.

SEQUENCE

Une séquence est une liste d'instructions permettant d'effectuer une action complexe.

Chaque instruction est séparée des autres par un ; (point virgule).

Une séquence débute par une {
Une séquence se termine par une }

break permet de quitter une séquence avant d'atteindre sa fin.

Chaque fonction comporte au moins une séquence.

On peut déclarer des variables dans une séquence, elles sont locales à la séquence.

VARIABLES DIMENSIONNEES…

C'est un regroupement de variables du même type.
Syntaxe :

```
type Nom_Variable [ nbre_elements ] ;
```

Accès aux données via un index.
L'index débute à 0 !

l'utilisation de typedef permet de rendre le code plus lisible et de partiellement contrôler les dimensions.

Création d'un nouveau type de variable dimensionnée :

```
typedef type_elements Nouveau_type[ nbre ] ;
```

Exemple :
* création d'un type appelé vect de 3 int.

```
typedef int vect [ 3 ] ;
```

* création d'une variable de type vect...

```
vect a ;
```

a est une variable de type vect, donc il existe a[0], a[1] et a[2].

On initialise les variables dimensionnées à l'aide d'une liste entre accolades :

```
vect a={ 0, 2, 4 } // a[0]=0, a[1]=2, a[2]=3
vect a={ 0 }; // a[0] ainsi que tous les
```

Tableau a deux dimensions…

Deux index : deux jeux de crochets !

Déclaration directe :

```
int a[3][3] = { {1,2,3},{0} }
```

Tout ce qui n'est pas explicitement initialisé est mis à 0 par le compilateur.

Déclaration avec type :
typedef int mat[3][3] ;
Ou

```
typedef int vec[ 3 ] ;
typedef vect mat[ 3 ] ;
```

Et l'on crée une variable :

```
mat a = { {1,2,3},{0} } ;
```

Accumulateur de sommation :

S = S +…

Permet de calculer une somme

Doit être initialisé à 0

Accumulateur de produit :

P = P*…

Permet de calculer un produit

Doit être initialisé à 1

Utilisent a priori une boucle for

Saisie dans un intervalle…

But contraindre la saisie, empêcher la saisie de valeurs non autorisées.

Exemple : saisir des notes entières entre 0 et 20

```
Faire
      saisir une note
Tant que ( note > 20 OU note <0 ) ;
```

Transtypage…

Permet de changer le type d'une variable localement.

Exemple :

```
int a ;
(float)a  = 3.1415 ;
printf("%f\n", (float)a ) ;
```

Remarques Opérateurs…

En C l'opérateur s'adapte aux opérandes.

Ainsi si :

```
int a=3 , b = 2 ;
float c= 0 ;

c = a/b ; // c contient 1 : division entière
c = (float)a/b // c contient 1.5
```

TODO

Ecrivez un programme C qui saisisse n notes (n défini a l'aide d'un define, les affiche, calcule et affiche la moyenne.

On définira un type tablo, de n variables entières.

La moyenne sera affichée avec 3 décimales.