Travaux pratiques n°1 (2h):
Introduction au langage C

Objectifs

- Introduction au langage C
- Coder des algorithmes détaillés en langage C
- Découvrir les premières erreurs de compilation

Partie 1: Introduction pratique au langage C

Pour passer du programme *C* (ou *programme source*) à un programme exécutable opérez de la façon suivante depuis un terminal :
1 – compilation et édition des liens : *gcc essai.c –o essai* ↵
2 - exécution : *./essai* ↵

Note : par simplicité nous emploierons dans la suite de ce texte le terme *compiler*, pour désigner les 2 étapes de compilation et d'édition des liens.

Entrées-Sorties 1

On appelle les entrées-sorties les communications entre un programme et l'utilisateur de ce programme.

Les entrées

L'instruction qui gère les *entrées*, c'est à dire la saisie de valeurs fournies par l'utilisateur à l'intention du programme est **scanf** dont l'équivalent dans le langage algorithmique est **demander**. Chaque valeur saisie par **scanf** doit être stockée dans une variable du programme.
NOTE IMPORTANTE : Comme il le sera précisé en cours, dans un **scanf** il faut obligatoirement précéder le nom de la variable avec le caractère **&**. On vous expliquera à quoi correspond ce **&** ultérieurement.

Les sorties

Comme nous l'avons vu l'instruction qui gère les *sorties,* c'est à dire l'affichage de valeurs fournies par le programme vers l'utilisateur, est **printf** dont l'équivalent dans le langage algorithmique est **afficher**.

De façon générale une instruction **printf** peut afficher plusieurs valeurs, certaines de ces valeurs étant le résultat d'une expression (rappel : une expression est la description d'une opération qui rend un résultat comme, par exemple, une opération arithmétique).

Utiliser votre éditeur favori pour créer le programme suivant (inclure les commentaires).

#include <stdio.h>

```
int main(void)
{               /* DEBUT DES DECLARATIONS */

                /* variables */
        int   a  =  5   ;
        float b  = 6.3 ;


        /* DEBUT DES INSTRUCTIONS */

        printf("\n debut du programme" ) ;

                /* affichage des resultats*/
        printf("\n a vaut : %d", a +1 ) ;  /* %d est le format pour une variable de type int */
        printf("\n b vaut : %f",  b * 2)   ; /* %f est le format pour une variable de type float
*/

        printf("\n fin du programme" ) ;
        return (0) ;
}
```

Compilez et exécutez le.

Puis modifiez le pour que les valeurs de a et de b soient demandées à l'utilisateur.

Entrées-Sorties 2

Vous trouverez ci-dessous quelques précisions sur **scanf** et **printf** qui vous seront nécessaires pour affiner vos programmes:

printf(format,listevaleurs) affiche la liste de valeurs (variables ou expressions) dans le format choisi. Le format est une chaîne de caractères entre guillemets (double quote "), comprenant un texte qui sera écrit tel quel, des spécifications de format (débutant par %) et des caractères spéciaux (\). A l'exécution, chaque spécification de format sera remplacée par une valeur de liste valeurs (dans l'ordre d'apparition des formats).

Une spécification de format est de la forme :

%[largeur][.précision]type (entre [] facultatifs)

La **largeur** est le nombre minimal de caractères à écrire (des blancs sont rajoutés si nécessaire). Si le texte à écrire est plus long, il est néanmoins écrit en totalité. Attention dans le cas d'un réel le point (.) est compté.

La **précision** définit, pour les réels, le nombre de chiffres après la virgule (doit être inférieur à la largeur). Dans la cas d'entiers, indique le nombre minimal de chiffes désiré (ajout de 0 sinon).

Le **type** est : c (char), s (chaîne de caractères, jusqu'au \0), d (int), u (entier non signé), x ou X (entier affiché en hexadécimal), o (entier affiché en octal), f (réel en virgule fixe), e ou E (réel en notation exponentielle), g ou G (réel en f si possible, e sinon), p (pointeur), % (pour afficher le signe %).

Les caractères spéciaux utilisables dans le format sont \t (tabulation), \n (retour à la ligne), \\ (signe \).

exemple:

printf("moyenne calculée :%5.2f\n",moyenne);

Affiche la moyenne sur 5 chiffres minimums avec 2 chiffres après la virgule (ex: 12.45 ou 2.50)

scanf(format,listeadresse) lecture au clavier de valeurs, dans le format spécifié. Les arguments sont les adresses des variables résultats. **scanf** retourne le nombre de valeurs effectivement lues et mémorisées.

Le casting: introduction

Il est possible que lors d'un calcul les opérandes n'ont pas le même type ou que le type du résultat de l'opération que vous souhaitez obtenir n'est pas celui des opérandes. Il faut donc expliciter le type désiré par la syntaxe suivante: *(type) expression*.

Ainsi, par exemple supposons la division de 1 par 2. Par défaut, le langage C effectuera une division entière arrondie à l'entier inférieur. Ainsi 1/2 donne comme résultat 0. Si vous voulez obtenir le résultat sous forme d'un float, vous devez écrire:
(float) 1/2;

exemple:

```
#include <stdio.h>
int main(void)
{
 float  un;
 int n=2;

 un = 1;
          un= un + (float) 1/(n*n);
 printf("\n un = %f", un);
 return(1);
}
```

va afficher:
un = 1.25

Alors que
#include <stdio.h>

```
int main(void)
{
 float  un;
 int n=2;

 un = 1;
 un= un + 1/(n*n);
 printf("\n un = %f", un);
 return(1);
}
```

va afficher:

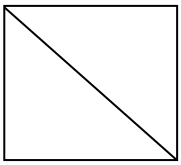
un = 1

Partie 2: Premiers programmes en langage C

Exercice 1

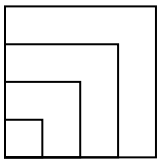
On dispose d'une tortue qui est capable de dessiner des traits sur l'écran de votre ordinateur. Vous disposez de 2 ordres possibles: *avance(longueur)* qui a pour résultat le tracé d'un segment de droite sur l'écran à partir de l'endroit où se trouve la tortue d'une certaine longueur (en points) et *tourne(angle)* qui fait tourner la tortue d'un certain angle (si l'angle est positif la tortue tourne à droite et à gauche si l'angle est négatif).

Écrire le programme en C (sans itérative) qui permet à la tortue de dessiner la figure ci-dessous (un carré de longueur 40 points) sans jamais repasser sur un segment déjà tracé.



Exercice 2

Donnez le programme en C (sans itérative) qui fera dessiner à la tortue la figure suivante:
La taille du côté du carré le plus petit est de 10 points.



Exercice 3

On veut peindre une pièce rectangulaire (excepté le plafond et le sol) dont les dimensions sont les suivantes :
longueur = *6*m, largeur =*5*m, hauteur = *2,5*m.
Cette pièce a une fenêtre carrée de *1.2* m de côté, l'épaisseur entre le cadre de la fenêtre (à peindre également) et la vitre étant de *15* cm.

On décide d'utiliser une peinture dont le pouvoir couvrant est de $3m^2$/litre. Ces peintures sont vendues par pot de 2 litres.
Sachant qu'on peint les murs et le cadre de la fenêtre calculer et afficher la quantité de peinture nécessaire ainsi que le nombre de pots nécessaires.

NOTE pour l'utilisation de la tortue.

soit un programme *essai.c* utilisant la tortue,vous devez:

1. Ajouter la directive d'inclusion :
   #include <Tortue/Tortue.h>
   ou

#include "Tortue.h" si vous utilisez la librairie dans votre répertoire courant

3. Séparer les étapes de compilation et d'édition des liens en tapant:

   1) compilation :
   gcc –c essai.c –o essai.o
   2) édition des liens :
   gcc –o essai essai.o -lTortue

   ou taper
   make essai si vous êtes avec la version locale.

Exemple:

/* ….une entête décrivant des informations sur le contenu du fichier, une description du problème qu'il résout, les différentes versions, les dates de création et de modification, les auteurs, etc ... */

```
/* DIRECTIVES D'INCLUSION *
#include <stdio.h>
#include  <Tortue/Tortue.h> // cf.  autre version du include si version locale

int main(void)
{
                        /* DEBUT DES INSTRUCTIONS */
 printf("\n debut du programme essaiTortue" ) ;
 printf("\n Observez ce que sait faire la tortue");                    */
 ouvre();        /* ouvre une fenêtre graphique, jamais de printf ou de scanf
                        après cette instruction */
 avance(100);
 tourne(90);
 avance(100);
 ferme();        /* ferme le fenêtre graphique, scanf et printf à nouveau autorisés */
 printf("\n fin du programme essai" ) ;
 return (0) ;
 }
```