

Travaux Pratiques N°1

R107

Avant-propos

Vous avez appris, dans la ressource R108 à manipuler des fichiers en ligne de commandes. Il est impératif que vous continuiez à utiliser ces commandes en lignes lors de vos séances de travaux pratiques de programmation en langage C.

Dorénavant, vous associerez à chaque séance de Travaux Pratiques, un sous-répertoire portant le nom : `~/R107/TP x` où x correspond au numéro du TP. Ce répertoire devra contenir le ou les fichiers de programmes écrits au cours de la séance ainsi que le **Makefile**¹ qui permet de les compiler et le compte-rendu que vous imprimez en fin de séance.

Pour chacun des programmes à écrire, il est impératif auparavant d'écrire un algorithme en pseudo-langage qui devra figurer dans votre compte-rendu (en .txt ou .pdf). Outre vos algorithmes, le compte-rendu expliquera vos choix (si besoin) et donnera des exemples d'exécution. Chaque programme sera également commenté. Tout ce que vous produirez en TP (compte-rendu et codes sources) sera autorisé le jour du DS. A l'issue de chaque séance, vous devez soumettre sur Moodle (dans l'espace de cours RT1R07) un fichier d'archive **.tar** contenant l'intégralité du répertoire **TP x** et dont le nom (sans extension) correspond à *nomdefamille_TP x* (par exemple l'étudiant **Gilles Rousselle** soumettant le TP N°1 déposera le fichier **rouselle_TP1.tar**). Seule la dernière soumission de votre TP sera corrigée!

Tout manque de respect à ces consignes se traduira par des points en moins.

Exercice 1

- Traduisez en langage C l'algorithme en pseudo-langage suivant, utilisez **gedit** pour éditer ce fichier (vous pouvez préférer **vi** ou **nano** à vos risques et périls, mais vous n'utiliserez pas d'IDE tel que **Code::Blocks** ou **Eclipse**), vous appellerez votre fichier **premier.c** :

Programme premier
Declaration

1. Voir plus loin dans le texte.

```

constante anneecourante en Entier
variable anneenaissance en Entier
variable age en Entier
Fin declaration
Debut
  afficher ("Quelle est votre année de naissance : ")
  lire(anneenaissance)
  age ← anneecourante - anneenaissance
  afficher("Vous avez ")
  afficher(age)
  afficher(" ans.")
Fin

```

- À l'aide du compilateur **gcc** et de ses options, compilez le programme précédemment écrit en lui donnant comme nom d'exécutable **premier**. Exécutez le. Notez le mode opératoire.
- La méthode la plus souvent utilisée pour compiler des programmes sous UNIX est l'utilisation d'un fichier **Makefile** et de l'utilitaire **make**. Particulièrement intéressante lors de compilations de plusieurs fichiers, cette façon de procéder permet de gagner du temps à la compilation, en ne compilant que les fichiers qui ont été modifiés depuis la dernière compilation.

Dans un fichier que vous sauvegarderez sous le nom de **Makefile** écrivez les lignes suivantes :

```

CC=gcc
FLAGS= -Wall
LIBS= -lm          # utile si vous utilisez <math.h>
EXEC= premier
%.o: %.c
    ↳ $(CC) $(FLAGS) -c $<

all: $(EXEC)

premier: premier.o
    ↳ $(CC) $(LIBS) $(FLAGS) -o $@ $^

```

clean:

`└─> rm *.o`

où le symbole `└─>` correspond à une tabulation (vérifiez dans les préférences de **gedit** que les tabulations ne sont pas remplacées par des espaces). Commentez ce **Makefile** en vous aidant de "l'introduction à Makefile" disponible sur e-campus ainsi que dans votre polycopié (également disponible sur e-campus).

Compilez en tapant **make** (ou **make premier**, ou **make all**). Remarquez qu'il y a plus de messages que lors de la compilation sans **Makefile**, en particulier vous devriez voir apparaître :

In function 'main':

attention : contrôle a atteint la fin non void de la fonction

Trouvez une solution pour que ce message n'apparaisse plus.

Exercice 2

1. Écrivez en pseudo-langage l'algorithme du programme **emprunt** qui calcule la mensualité d'un crédit à taux fixe. Il demandera le montant du capital emprunté, le nombre d'années et le taux annuel. La formule de calcul est la suivante :

$$M = C \times T \times \frac{(1 + T)^N}{(1 + T)^N - 1}$$

où M est la mensualité, C le capital emprunté, T le taux mensuel (c'est-à-dire le taux annuel / 12) et N le nombre de mois (nombre d'années / 12). Le taux est donné en pourcentage (par exemple **5.5** pour un taux à 5,5%).

2. Traduisez cet algorithme en Langage C.
Pour élever un nombre à la puissance d'un autre, vous pourrez utiliser la fonction **pow()** de la bibliothèque mathématique **<math.h>**. Ainsi **pow(x,y) = x^y** .
3. Sachant que l'iPhone 13 Pro coute 1159€ et que le taux d'un prêt à la consommation serait de l'ordre de 4,79%, quelles seraient vos mensualités si vous vous engagez sur deux ans ?
4. Modifiez le **makefile** pour compiler également le programme **emprunt** et tel

que la commande `make emprunt` compile le programme `emprunt` mais également que la commande `make all` compile à la fois `premier` et `emprunt`.