Licence Informatique — **Initiation à la Programmation**
Université Bordeaux I — **Travaux Dirigés**

Environnement de programmation

- Expérimenter les outils de programmation make, emacs, gcc, indent.
- Respecter un style de programmation.
- Écrire un programme principal : utiliser les arguments du main et définir une fonction d'usage.

Un compte-rendu de l'exercice 3 vous sera demandé la semaine prochaine.

Exercice 1.

PRÉPARATION

1. sortir le necéssaire pour prendre des notes ;
2. se connecter, en cas de problème contacter très rapidement votre enseignant ;
3. recopier à votre racine le répertoire `~wacren/IP` ainsi que son contenu (`cp -r ~wacren/IP ~`) ;
4. visiter succintement votre répertoire `~/IP` à l'aide des commandes `cd`, `ls` et `cat` ;

ÉDITION D'UN FICHIER

5. lancer emacs en arrière plan[1] (`emacs &`).
6. sous emacs charger le fichier `~/IP/1_envi_prog/Pi/pi.c` en tapant
   `C-x C-f`[2] `~/IP/1_envi_prog/Pi/pi.c`;

COMPILATION

7. sous un shell, vous rendre au répertoire `Pi` (`cd  ~/IP/1_envi_prog/Pi`), lister son contenu, puis exécuter les commandes suivantes :
```
gcc -c pi.c -o un_point_o.o
ls
gcc  un_point_o.o -o un_executable
ls
gcc  un_point_o.o -lm -o un_executable
ls
un_executable
rm  un_executable un_point_o.o
```
8. compiler sous emacs le programme `pi.c` pour cela taper sur la touche "F9" puis valider (i.e. taper sur la touche "Entrée") la commande `make -k` qui lance l'interprétation du fichier `Makefile` ;
9. analyser le compte-rendu de la compilation (la séquence de touches `C-x 1` permet de fusionner les fenêtres d'emacs, `C-x 0` pour tuer la fenêtre courante, `C-x o` pour changer de fenêtre) ;
10. analyser le fichier `Makefile` ;
11. dans le shell, exécuter le programme pi puis nettoyer le répertoire courant en tapant `make clean`, puis lister le répertoire courant ;

[1] il est inutile voire problématique de lancer plusieurs processus emacs en parallèle

[2] ça se dit "contrôle x, contrôle f" et ça signifie appuyer simultanément sur la touche "Ctrl" et sur la touche "x" puis appuyer simultanément sur "Ctrl" et "f"

Standardisation de la présentation

12. sous emacs, améliorer la présentation du buffer pi.c (en tapant C-x b pi.c au besoin) , on pourra utiliser la touche “tab” afin d’indenter plus facilement le texte ;
13. sous emacs, copier le buffer pi.c dans le fichier mon_pi.c en tapant d’abord C-x C-w puis en entrant le nom du fichier ;
14. sous le shell, taper la commande indent pi.c ;
15. sous emacs charger pi.c puis comparer les buffers en les affichant simultanément : découper la fenêtre verticalement C-x 3, puis visualiser le buffer mon_pi.c en tapant C-x b puis mo puis “Tab” pour accélérer la frappe ;

Compilation séparée

16. parcourir les fichiers du répertoire ~/IP/1_envi_prog/Devise;
17. sous emacs supprimer le buffer mon_pi.c (C-x k mon_pi.c C-x 0) puis créer le fichier nombre.c (C-x 2 C-x C-f nombre.c), y déplacer la ligne #include <math.h> (se placer au début de la ligne pour taper C-k C-x o C-y) puis la fonction calculant π grâce une série de C-k par exemple ou mieux en utilisant la notion de bloc (marquer une extrémité C-*espace*, se déplacer à l’autre extrémité, découper le texte C-w puis changer de fenêtre et coller le texte C-x o C-y) ;
18. à l’image des fichiers du répertoire ~/IP/1_envi_prog/Devise/ créer le fichier nombre.h et modifier les fichiers Makefile et pi.c afin que ce dernier ne comporte que le programme principal ;
19. observer le résultat produit par gcc -MM *.c ;
20. compiler votre programme (nettoyer via make clean avant chaque nouvelle compilation).

Exercice 2. Commenter l’exécution des programmes ~/IP/1_envi_prog/Arguments/Hal.c et ~/IP/1_envi_prog/Arguments/Combien.c.

Écrire un programme qui liste ses arguments en les numérotant.

Exercice 3.

1. compiler le programme ~/IP/1_envi_prog/Exponentiel/main.c, traduire en français les messages d’erreurs ;
2. améliorer le programme (correction et lisibilité) ;
3. modifier la fonction main() pour que l’utilisateur puisse choisir la précision du calcul dans la ligne de commande ;
4. écrire une fonction d’usage (voir le fichier convertir_en.c) ;
5. intégrer la fonction exponentiel au module nombre.c et utiliser ce module ;
6. imprimer au format “2 pages / feuille” (voir la commande a2ps) le résultat de votre travail (makefile, fichiers en-têtes et sources, éventuels commentaires)