
TP 7 VARI 1

Informations techniques PC Suse :

- (a) Pour démarrer une session : utilisateur `licencep` et mot de passe `7002n**`.
 - (b) Pour démarrer *Processing* : clic sur la tête de coméléon en haut à droite → Développement → Processing.
 - (c) La page VARI1 : `cedric.cnam.fr/~porumbed/vari1/`
 - (d) Pour démarrer un terminal : tête de caméléon → Terminal → Konsole.
-

1 Commandes réseaux

Démarrer une console/terminal grâce à la combinaison de touches : CTRL gauche ⊕ ALT gauche ⊕ t. Si cela ne fonctionne pas, voir le point (d) plus haut.

Exercice 1 Taper dans ce terminal la commande `echo "Salut toto"` et regardez la sortie.

Exercice 2 Taper les commandes suivantes une par une. On observe que la première commande indique l'OS plus précisément. Dérouler le résultat de la deuxième commande pour trouver combien de coeurs possède le processeur (CPU).

```
cat /proc/version
cat /proc/cpuinfo
```

Exercice 3 Lancer `ifconfig` et noter votre adresse IP et l'adresse de diffusion (broadcast) de la première interface. Au cnam, votre adresse IP commencera par 163.173. Vous pouvez aussi `ip addr` pour le même objectif.

Exercice 4 Taper la commande `route -n`. Quelle est votre passerelle (par défaut) qui permet de vous relier à l'Internet? Exécuter un `ping` sur la passerelle, c.à.d., taper `ping IP_PASSERELLE` (à arrêter avec CTRL-C).

Exercice 5 Taper `ping canterbury.ac.nz` et observez qu'on vous affiche l'adresse IP du site web de l'université de Canterbury (nl : Nouvelle Zélande). Le serveur web de cette université vous renvoie des réponses en boucle sans arrêt.

Exercice 6 Chercher dans le manuel l'objectif de la commande `ping` (taper "`man ping`", q pour sortir). Modifier la commande `ping` de l'exercice précédent pour la faire recevoir que 3 réponses.

Rappel : Pour chercher un mot clé comme `count` dans le manuel (de la commande `ping`), taper `/`, suivie `count` et d'*Entrée*. Ensuite, taper `n` pour rechercher l'apparition suivante du mot.

Exercice 7 On souhaite utiliser la commande `traceroute` pour nous aider à localiser les serveurs traversés pour joindre `canterbury.ac.nz`. Par contre, il faut taper `/usr/sbin/traceroute canterbury.ac.nz` pour exécuter `traceroute`. On peut toujours consulter un "IP Locator" sur internet, voir `www.iplocation.net`. Si vous êtes sous windows, vous pouvez taper `tracert` à la place de `traceroute` dans l'invité de commande Windows.

Exercice 8 Quel est le résultat de la commande ci-dessous.

```
for((i=0;i<7;i++)) do echo toto; done
```

Modifiez cette commande pour la faire afficher «toto» 15 fois.

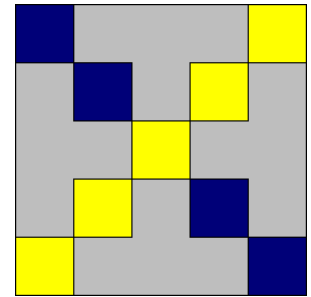
Exercice 9 (bonus) Taper `netcat -v cedric.cnam.fr 80` pour se connecter par TCP au port 80 (web) du site web `cedric.cnam.fr`. Taper un mot au hasard lorsque la connexion est établie. Remarquer la réponse envoyée par le site web `cedric.cnam.fr`.

Exercice 10 Tapez `pwd`. Qu'affiche cette commande selon son manuel (à consulter) ?

2 Processing

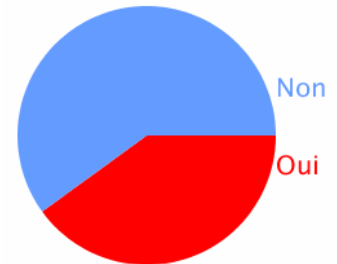
Exercice 1 Soit le code ci-dessous. Ajouter la deuxième diagonale composée de petits carrés jaunes pour faire le dessin en bas à droite. Pour obtenir la couleur jaune, on utilise une combinaison de rouge et vert !

```
1 size (250,250);
2 fill (190,190,190); //mélange rouge, vert, bleu=>gris
3 rect (0,0,250,250);
4 fill (0,0,120); //un peu de bleu 120
5 rect (0,0,50,50);
6 rect (50,50,50,50);
7 rect (100,100,50,50);
8 rect (150,150,50,50);
9 rect (200,200,50,50);
```



Exercice 2 L'objectif de cet exercice est de réaliser un diagramme camembert comme dans l'image à droite, qui pourrait par exemple correspondre aux résultats d'un référendum. Le pourcentages de votes «Oui» et «non» sont initialisés au hasard à l'aide du code ci-après. **Note** : utiliser `arc(...)`.

```
float oui = random(100);
float non = 100-oui;
```



Exercice 3 Déclarer une variable réelle x et affecter la valeur 2^8 à x , à l'aide de la fonction `pow(...)`. Par exemple, `pow(3,4)` renvoie 81. Afficher la valeur de x . Par la suite, initialiser une variable entière y à une valeur entière aléatoire entre 1 et 20, comme à la première ligne du programme de l'exercice suivant. Afficher la valeur de y et le résultat du calcul 2^y . Exécuter le programme plusieurs fois pour voir plusieurs puissances de 2.

Exercice 4 (bonus) Modifier le programme précédent pour réaliser les mêmes calculs sans fait appel à `pow`. Vous devriez utiliser une boucle `for` pour calculer toutes les puissances.

Exercice 5 Corriger les erreurs de compilation du programme à droite et exécuter le.

Exercice 6 Ajouter une variable $tp2$ initialisée au début du programme de la même manière. Si la note d'examen est inférieure à 7, afficher "échoué". Sinon, on calcule la note finale qui est définie comme la moyenne entre l'examen et la note de TP la plus grande. Exemple : si $exam = 10$, $tp1 = 5$, $tp2 = 12$, on ignore $tp1$ et on obtient $\frac{exam+tp2}{2} = 11$. Si la note finale est supérieure à 10, afficher "réussi", sinon afficher "échoué".

```
int exam = (int)random(20); //note exam
int tp1 = (int)random(20); //note TP
if (exam<7){
    println("Échec à l'examen");
}
if (exam=20){
    println("Vous êtes un génie");
} else
    println("Note < 20")
int note finale = exam+tp1)/2;
println("note finale=" noteFinale);
```

Exercice 7 Soit les variables $exam$, $tp1$, $tp2$, $tp3$ et $tp4$ initialisées au début du programme pour indiquer une note d'examen et resp. 4 notes de TP. Une note d'examen inférieure à 7 est éliminatoire. La plus petite note des quatre notes de TP est ignorée. La note finale est calculée ainsi : l'examen compte pour 60% et la moyenne des trois notes de TP restantes pour 40%. Exemple : si $exam = 10$, $tp1 = 5$, $tp2 = tp3 = tp4 = 12$, on obtient $10 \cdot 0.6 + 12 \cdot 0.4 = 10.8$. Si la note finale est supérieure à 10, afficher "réussi", sinon afficher "échoué".

Exercice 8 (bonus) Réaliser l'exercice précédent en utilisant un tableau de 4 notes de TP, ainsi qu'une boucle `for` pour calculer la plus petite note.

Exercice 9 (bonus) Écrire un programme qui trace un cercle de rayon 50 à chaque appel de `draw()`. Enregistrer les coordonnées x et y des cercles dans deux tableau de taille 100. **Ne tracez pas un nouveau cercle s'il touche un cercle existant**. Si jamais on arrive au cercle numéro 100, appelez `noLoop()`.