TP4 - boucles for, while (bis)

Laure Gonnord
`http://www-verimag.imag.fr/~gonnord/`

1 Notes et moyennes d'étudiants

Pour chacun des exercices on fera un fichier `.c` séparé, quitte à faire des copier-coller judicieux. AVANT de se mettre à coder, on lira tout l'énoncé, puis pour chaque exercice on commencera par faire un plan sur papier du programme à faire.

Exo 1

Écrire un programme qui calcule la moyenne de 4 notes d'un étudiant : Chimie, Physique, Mathématiques et Informatique. Le programme demande les 4 notes (entières) à l'utilisateur, vérifie qu'elles sont entre 0 et 20, dans ce cas calcule et affiche la moyenne (qui sera un `double`). Si une note n'est pas comprise entre 0 et 20, le programme affiche "INCORRECT".

Exo 2

Même chose sauf que la moyenne est pondérée de la façon suivante :
- Chimie : 3
- Physique : 4
- Mathématiques : 2
- Informatique : 2

Exo 3

On veut en plus de l'exo2 rajouter la mention de l'étudiant. On aura intérêt à rajouter une procédure `void affichementionn(int moyenne)`, qui étant donnée la moyenne affiche la mention de l'étudiant. On supposera dans cette procédure (qui ne retourne AUCUN résultat, donc ne comporte pas d'instruction `return`), que la moyenne est entre 0 et 20, et on ne fera qu'afficher la mention. En particulier, cette procédure ne fera aucun `scanf`. On fera un appel à cette procédure dans la fonction `main`, APRES avoir calculé ladite moyenne.

Exo 4

Cette fois on veut générer automatiquement les notes de l'élève. Pour cela, on va écrire une fonction `int genere_note(void)` qui renvoie un entier entre 0 et 20 :

```
int genere_note(void)
{
  int rando = rand();
  return (rando%21);
}
```

Ensuite, cette fonction peut être utilisée de la façon suivante : si `x` existe, `x=genere_note()` (remarquer l'absence d'argument) met une note entre 0 et 20 dans la variable `x`.

Dans le corps de la fonction principale, on commencera par générer les 4 notes voulues, puis on calculera et affichera la moyenne, enfin on affichera la mention de l'élève.

Exo 5

On veut comparer la moyenne de 2 étudiants. On commencera par générer les 8 notes, puis on calculera les 2 moyennes. Enfin, on affichera notes, moyenne et mention de l'étudiant le plus fort (avec la plus grosse moyenne), et ensuite celles de l'étudiant le plus faible.

On remarquera que si on voulait gérer 15 étudiants, on aurait besoin de $15 * 4$ variables, ce qui est beaucoup. On verra dans pas longtemps une façon élégante de résoudre ce problème.

Exo6

On veut gérer un nombre d'étudiants arbitraire. Pour chaque étudiant (désigné par son numéro : 1, 2 . . .) le programme générera les notes, calculera la moyenne, affichera moyenne et mention. L'utilisateur décide à chaque nouvel étudiant de s'arrêter ou pas (réponse 1 ou 0 à la question : "Voulez-vous continuer ?"). On utilisera une boucle while. On affichera à la fin la meilleure moyenne, ainsi que le numéro de l'étudiant qui a la meilleure moyenne.

2 Divisions en régions

Cet exercice est plus difficile (et volontairement moins dirigé) que le précédent. On fera moult dessins et essais de programme sur papier avant de se jeter sur l'ordinateur.

2.1 Dessin des régions

Écrire un programme qui demande à l'utilisateur d'entrer au clavier deux points p_1 et p_2 de coordonnées (x_1, y_1) et (x_2, y_2). Si ces deux points sont sur la même ligne horizontale (ordonnées égales), ou si ils sont sur la même ligne verticale (abscisses égales), le programme signale l'erreur et s'arrête. Sinon, il sépare le plan de la façon suivante :

```
          1          |           2            |    3
---------------------o------------------------|----------------
          4          |           5            |    6
---------------------|------------------------*----------------
          7          |           8            |    9
```

(le symbole `o` désigne p_1, le symbole `*` désigne p_2).

2.2 Un point dans une région

Ensuite, on demande un troisième point. Modifier l'affichage pour situer le point dans la région à laquelle il appartient :

```
          1          |           2            |    3
---------------------o------------------------|----------------
          4          |           5            |   [6]
---------------------|------------------------*----------------
          7          |           8            |    9
```

On passera un certain temps à caractériser chaque région de la sorte : x appartient à la région 6 si et seulement si . . .