

UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 159

Devoir #2 — Automne 2009

Programmation

Devoir à remettre au plus tard le 15 octobre 2009 à 23h59. La remise devra être faite grâce à la commande *turnin* (voir à la fin de l'énoncé pour plus de détails).

Le but de ce travail consiste à implanter la solution d'un problème dont vous recevez la spécification, l'analyse et la conception. L'objectif est de vous familiariser, au niveau programmation, avec la notion de séquence, de fonction, de sélection et d'itération. **Vous devez obligatoirement implanter la conception fournie en annexe en utilisant seulement les notions de programmation vues dans le cours.**

Vous devez me remettre votre programme, sa documentation, trois fichiers de tests ainsi que le formulaire d'estimation de l'effort. Votre programme devra respecter les normes de programmation du département. Les fichiers tests devront contenir des tests significatifs. La documentation se fera sous forme de commentaires dans le programme en utilisant les normes de *doxygen*.

Sujet : Tests de capacité physique

Lors de son camp annuel d'évaluation des recrues, un club de hockey professionnel désire évaluer ses nouvelles recrues. Pour y parvenir, il souhaite leur faire passer une série de tests afin de déterminer leur capacité physique. Vous devez leur fournir un programme général qui, à partir des résultats des tests, évalue la qualité des recrues (sous forme de notes sur 100 et de cotes) et détermine aussi la moyenne des résultats obtenus. Ce programme traite les évaluations d'une seule équipe à la fois mais peut être utilisé par plusieurs équipes sportives. Les différentes épreuves sportives entrant dans l'évaluation de la capacité physique d'une recrue sont :

- de manière obligatoire pour tous les joueurs
 - la capacité cardio-vasculaire ;
 - la vitesse sur patin sur un trajet précis ;
 - la précision des tirs ;
- de manière optionnelle
 - la force qui comprend au maximum dix exercices différents tels les redressements assis, les pompes, la levée de poids, les sauts, ...

- la souplesse qui comprend au maximum six exercices tels les exercices d'évaluation pour les épaules, les ischiojambiers, le bas du dos, ...

Les résultats des tests sont entrés sous forme de notes numériques. Comme le programme peut être utilisé par plusieurs équipes, on devra à chaque exécution spécifier les pondérations et les notes maximales associées aux différents tests. Pour les tests de force et de souplesse, on devra spécifier le nombre d'exercices effectués. Il est à noter que tous les exercices de force, ainsi que ceux de souplesse, ont la même note maximale et le même poids.

Les divers résultats obtenus par une recrue peuvent être fournis dans un ordre qui peut différer d'une recrue à l'autre. Les résultats d'un même test de force ou de souplesse sont cependant regroupés. Chaque résultat ou regroupement de résultats devra donc être bien identifié avant d'être entré. Les codes suivants servent à identifier les notes : «c» pour cardio-vasculaire, «v» pour vitesse, «p» pour précision, «f» pour force et «s» pour souplesse. Un identificateur spécifique indiquera que les résultats de cette recrue sont tous entrés. Avant de demander l'ensemble des résultats pour une recrue, vous devez afficher la liste des identificateurs de type d'évaluations afin de faciliter la tâche du préposé.

Finalement, une recrue est identifiée par un numéro. Un numéro spécial indiquera que l'entrée des résultats est terminée et que l'on peut calculer la moyenne.

Pour chaque exercice de force ou de souplesse, il est possible d'avoir une note négative indiquant que la recrue a été exemptée de cette activité. Cette note négative ne doit pas rentrer en ligne de compte pour le calcul du résultat final. Une recrue ne peut pas être exemptée de tous les exercices.

À la fin, le programme devra afficher une évaluation qualitative pour chaque recrue de la manière suivante :

- un résultat final supérieur ou égal à 90% donne «recrue exceptionnelle» ;
- un résultat final supérieur ou égal à 80% donne «excellente recrue» ;
- un résultat final supérieur ou égal à 70% donne «très bonne recrue» ;
- un résultat final supérieur ou égal à 60% donne «bonne recrue» ;
- un résultat final supérieur ou égal à 50% donne «recrue médiocre» ;
- un résultat final inférieur à 50% donne «recrue rejetée».

Analyse globale du problème

Entrée :

- (clavier) pondération pour cardio, vitesse, précision (entiers)
- (clavier) pondération pour force et souplesse (entiers)
- (clavier) note maximale pour cardio, vitesse, précision (réels)
- (clavier) note maximale pour force et souplesse (réels)
- (clavier) nombre d'exercices de force et de souplesse (entiers)
- (clavier) numéro de la recrue (entiers)
- (clavier) codes de l'évaluation (caractère)
- (clavier) notes (réels)

Sortie :

- (écran) suite d'évaluation qualitative (chaîne de caractères)
- (écran) moyenne (réel)
- (écran) on peut optionnellement afficher d'autres informations tel, par exemple, le numéro de la recrue pour accompagner la note, afin d'avoir une meilleure présentation des résultats.

Formule :

moyenne = $\sum \text{notes} / \text{nombre de notes}$

note pondérée = $(\sum \text{notes} / \text{nbr notes}) * \text{pondération} / \text{max} * 100$

Constantes

bornes : 90, 80, 70, 60 et 50.

choix d'exercices : v (vitesse), c (cardio), p (précision), s (souplesse),
f (force) et x (fin)

Analyse complémentaire

Dans un premier temps, les informations suivantes sont demandées dans l'ordre ci-après :

- la pondération pour le cardio
- la note maximale pour le cardio
- la pondération pour la vitesse
- la note maximale pour la vitesse
- la pondération pour la précision
- la note maximale pour la précision
- le nombre d'exercices de force et, le cas échéant, :
 - la pondération de l'ensemble des exercices
 - la note maximale des exercices de force
- le nombre d'exercices de souplesse et, le cas échéant, :
 - la pondération des exercices de souplesse
 - la note maximale des exercices de souplesse

Ensuite, pour chaque recrue on demande d'abord son numéro (un numéro nul indique la fin du traitement) puis une liste de notes précédée pour la plupart d'un code. La signification des codes est le suivant :

- v ou V indique que ce qui suit est la note de la vitesse ;
- c ou C indique que ce qui suit est la note pour le cardio ;
- p ou P indique que ce qui suit est la note pour la précision ;
- s ou S indique que ce qui suit sont les notes pour la souplesse ;
- f ou F indique que ce qui suit sont les notes pour la force.

Enfin un code x ou X indique que l'on peut maintenant déterminer la note finale et la cote de cet étudiant.

Conception (Diagramme structurel et flots de données)

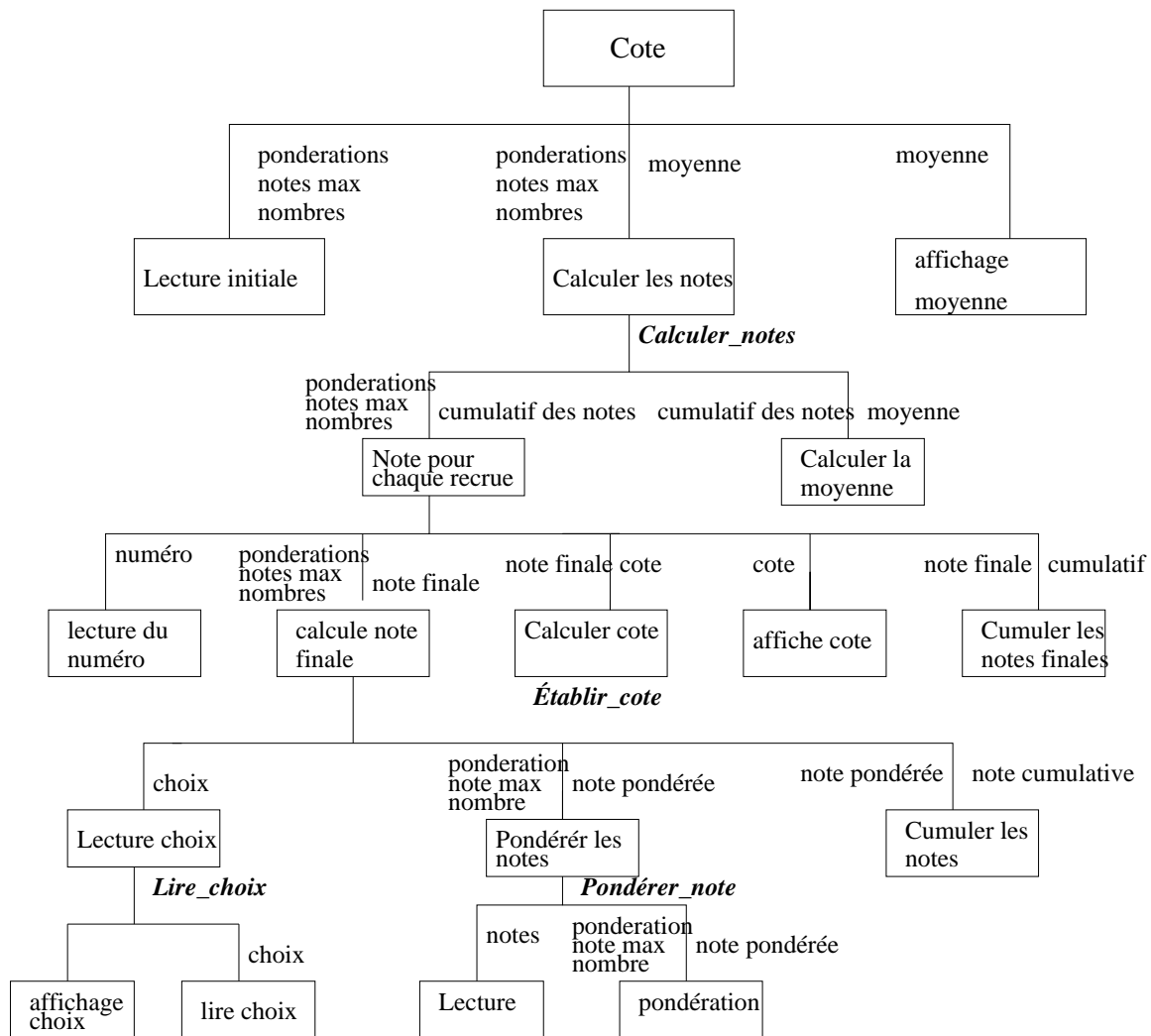


FIG. 1 – Diagramme structurel

MODULE *Évaluation Recrue*

Analyse

Entrée :

- (clavier) les pondérations pour force, vitesse, cardio et souplesse (entiers)
- (clavier) les notes maximales pour cardio, vitesse, force et souplesse (entiers)
- (clavier) le nombre de notes pour les exercices de force et de souplesse (entiers)

Sortie : (écran) la moyenne des notes (réel)

Conception (algorithme)

1. lecture
 - de la pondération pour le cardio
 - de la note maximale pour le cardio
 - de la pondération pour la vitesse
 - de la note maximale pour la vitesse
 - de la pondération pour la précision
 - de la note maximale pour la précision
 - du nombre d'exercices pour la force
 - s'il y a au moins un exercice lire
 - la pondération de l'ensemble des exercices de force
 - la note maximale des exercices de force
 - du nombre d'exercices de souplesse
 - s'il y a au moins un exercice lire
 - la pondération pour les exercices de souplesse
 - la note maximale pour les exercices de souplesse
2. calculer les notes (Module «Calculer notes»)
3. afficher moyenne des notes

MODULE «*Calculer notes*»

Analyse

Entrée :

- (paramètre) les pondérations pour cardio, vitesse et précision (entiers)
- (paramètre) les pondérations pour force et souplesse (entiers)
- (paramètre) les notes maximales pour cardio, vitesse et précision (entiers)
- (paramètre) les notes maximales pour force et souplesse (entiers)
- (paramètre) le nombre de notes pour les exercices de force et de souplesse (entiers)
- (clavier) les numéros des recrues (entiers)

Sortie :

- (écran) suite d'évaluation qualitative (chaînes de caractères)
- (retour) la moyenne des notes (réel)

Constantes : choix d'exercices : v (vitesse), c (cardio), p (précision), s (souplesse),
f (force) et x (fin)

Formule : moyenne = $\sum \text{notes} / \text{nombre de notes}$

Conception (algorithme)

1. pour chaque recrue (fin sur un numéro nul)
 - (a) lire son numéro
 - (b) pour chaque liste de notes (calcul de la note finale)
 - i. identifier le type des notes (module «Lire choix»)
 - ii. selon le choix (cardio, vitesse, précision, force, souplesse)
on traite les notes lues (module «Pondérer note»)
 - iii. cumuler les notes
 - (c) établir la cote (module «Établir cote»)
 - (d) afficher la cote
 - (e) cumuler les notes finales
 2. calculer la moyenne
-

MODULE «*Établir cote*»

Analyse

Entrée : (paramètre) note finale (réel)

Sortie : (retour) évaluation qualitative (chaîne de caractères)

Constantes : borne1 = 90, borne2 = 80, borne3 = 70, borne4 = 60, borne5 = 50

Conception (algorithme)

1. Établir la cote en fonction des bornes.

MODULE «*Lire choix*»

Analyse

Entrée : (clavier) le choix d'exercice (caractère)

Sortie :

- (écran) les différents choix d'exercices (caractères)
- (retour) le choix d'exercice (caractère)

Constantes : choix d'exercices : v (vitesse), c (cardio), p (précision), s (souplesse),
f (force) et x (fin)

Conception (algorithme)

1. afficher tous les choix possibles
 2. lire le choix
-

MODULE «*Pondérer note*»

Analyse

Entrée :

- (paramètre) nombre de notes (entier)
- (paramètre) pondération des notes (réel)
- (paramètre) note maximale (réel)
- (clavier) notes (réel)

Sortie : (retour) note pondérée (réel)

Formule : note pondérée = (somme notes / nbr notes) * (pondération / note max. * 100)

Conception (algorithme)

1. pour chaque note de l'évaluation
 - (a) lecture de la note
 - (b) si note ≥ 0
 - i. sommation des notes
 - ii. on compte le nombre de notes effectives pour la pondération
2. pondérer la note (formule)

Soumission

Vous devez soumettre votre travail pratique avant la date limite. Pour cela utiliser la commande

```
«turnin -cift159 -ptp3 tp3.cpp test1 test2 test3 calculEffort.txt»
```

pour soumettre :

- un fichier portant **OBLIGATOIREMENT** le nom *tp3.cpp* qui contient le programme demandé. Si nécessaire, vous pouvez utiliser plusieurs fichiers : les noms suggérés sont *tp3n.cpp* ou *n* variera de 1 au nombre nécessaire. Dans ce cas la commande de soumission sera
`turnin -cift159 -ptp3 test1 test2 test3 calculEffort.txt tp31.cpp tp32.cpp ...`
- trois fichiers tests portant **OBLIGATOIREMENT** les noms *test1*, *test2* et *test3*. Les trois tests soumis doivent être significatifs, i.e. qu'ils permettent de tester le plus de cas possibles pour votre application.
- un fichier portant **OBLIGATOIREMENT** le nom *calculEffort.txt* qui est le formulaire d'estimation de l'effort que vous pouvez vous procurer sur le site Web du cours. Pour générer un fichier texte, vous n'avez qu'à utiliser le fichier en format chiffré (.xls) et à copier/coller les données dans un fichier texte (.txt) par l'intermédiaire d'un éditeur. Vous pouvez aussi le sauver en format « .csv » et le renommer en format « .txt ».

Étant donné que la correction est partiellement automatisée, une mauvaise soumission entraîne une mauvaise note (possiblement 0). Il n'y a pas de recorection.

Vous pouvez aussi soumettre un seul **répertoire** portant **OBLIGATOIREMENT** le nom **tp3** qui contient le programme demandé (**tp3*.cpp**), les trois fichiers tests (**test1**, **test2**, **test3**) et le fichier pour le calcul de l'effort. Pour cela, vous devez vous placer en dehors du répertoire et faire la commande :

```
turnin -cift159 -ptp3 tp3
```

Pour vérifier si le programme a bien été soumis, vous pouvez faire la commande :

```
turnin -v -cift159 -ptp3
```

Avant de soumettre vos travaux, faites exécuter votre programme dans les mêmes conditions que celles du train de travaux. Pour cela mettre vos données dans un fichier (par exemple de nom *donnees.tp3*) et vos résultats dans un autre fichier (par exemple de nom *resultats.tp3*).

Les seuls «**#include**» valides sont des «**#include**» de fichiers de suffixe «.h». Il faut bannir les «**#include**» de fichiers ayant pour suffixe «.cpp».

Exemple pour le dernier point :

avec un éditeur de texte (nedit, textedit, kate, ...), créer le fichier *donnees.tp3*

```
g++ tp3.cpp -o tp3
```

```
tp3 < donnees.tp3 > resultats.tp3
```

```
less resultats.tp3
```