TD2 : Itération (2 séances)

# Simulation d’un traitement itératif

**Algorithme** Multiples

*{Saisir une suite de nombres entiers fournis par l’utilisateur, avec arrêt sur la valeur STOP, et indiquer à l'utilisateur si ces nombres sont multiples d'une constante donnée.}*

**constante** (STOP : **entier**)  9999

**(**MULT : **entier**)  7 **variables** uneVal, nbMult : **entiers début**

nbMult  0

**afficher** ("Entrer une suite de nombres entiers, ", STOP, "pour finir.")

{l’utilisateur saisit son premier numéro, on vérifie si ce n’est pas la valeur d’arrêt STOP puis on vérifie si la valeur est un multiple de MULT}

**saisir** (uneVal)

*{Saisie d'une suite de nombres, arrêt sur STOP}*

**tant que** uneVal  STOP **faire**

*{compter le nombre de multiples de MULT}*

**si** uneVal mod MULT = 0 nbMult  nbMult + 1

**fsi**

{on saisit à nouveau une valeur avant la répétition de la boucle.}

**saisir** (uneVal)

**ftq**

*{affichage du résultat}*

**afficher** ("Vous avez saisi", nbMult, "multiples de ", MULT)

**fin**

* Cet algorithme attend des données de la part de son utilisateur. Quels affichages obtenez-vous si vous lui fournissez les données 22, 49 et 9999 ?

Si on fournit les données 22, 49 et 9999, on obtient les affichages « Vous avez saisi 1multiples de 7 ».

* Ajouter les deux commentaires manquants afin d’identifier clairement les différentes parties du

traitement itératif effectué par la boucle **tant que**.

Voir l’algorithme .

* Expliquer pourquoi la boucle **tant que** est préférable ici à la boucle **répéter** et à la boucle **pour**.

Ici, la boucle tant que est préférable à la boucle pour car elle nous permet d’entrer autant de valeurs voulues tant que l’utilisateur ne veut pas s’arrêter. Il faudrait connaître le nombres de valeurs que l’on veut rentrer à l’avance pour utiliser une boucle pour.