

Travail pratique #1

Ce travail doit être fait individuellement

Notions mises en pratique : les algorithmes en pseudocode (affectations, lecture, écriture, sélections, boucles).

1. Travail à faire

Téléchargez le fichier TP1.txt (sur Moodle, dans la section CONTENU DU COURS / Travaux) et complétez-le en écrivant votre solution pour chacun des 3 exercices ci-dessous. C'est ce fichier complété que vous devez remettre.

Vous ne devez pas modifier le type du fichier à remettre. C'est un fichier texte avec l'extension .txt. Pour ouvrir le fichier TP1.txt, et le compléter, utilisez un éditeur de texte, par exemple :

Sur Windows : **Notepad++**
<https://notepad-plus-plus.org/download/>

Sur Mac OS X : **TextWrangler**
<http://www.barebones.com/products/textwrangler/download.html>

Le fichier TP1.txt est encodé en UTF-8, et cet encodage ne doit pas être modifié.

Rapportez-vous aux exemples d'exécution de chaque exercice pour voir les messages et informations à afficher à l'écran.

Appliquez-vous à indenter correctement votre pseudocode. Pour indenter vos algorithmes, utilisez une indentation de 3 ou 4 espaces (ou bien 3, ou bien 4... Restez constant !). Dans la plupart des éditeurs de texte, vous pouvez ajuster la tabulation au nombre d'espaces que vous désirez.

Dans le fichier TP1.txt, limitez la longueur des lignes à un maximum de 80 caractères.

Utilisez des constantes autant que possible.

Évitez le plus possible la répétition inutile des instructions, par exemple :

À éviter	À faire
<pre>si a > b alors afficher "allo" afficher "ok" sinon afficher "salut" afficher "ok" fin si</pre>	<pre>si a > b alors afficher "allo" sinon afficher "salut" fin si afficher "ok"</pre>

Note : Si quelque chose est ambigu, obscure, s'il manque de l'information, si vous ne comprenez pas les énoncés, si vous avez des doutes... vous avez la responsabilité de vous informer auprès de votre enseignante.

Exercice 1 (30 points)

Écrivez l'algorithme en pseudocode d'un programme qui calcule les points gagnés lorsqu'on lance 5 dés dans un jeu de dés imaginaire. L'algorithme affiche d'abord le message de présentation suivant : *"Ce programme calcule les points gagnés selon la somme de 5 des lances."* Ensuite, le programme demande à l'utilisateur d'entrer la somme des 5 dés qu'il a lancés. Une somme valide doit être un nombre entier entre 5 et 30 inclusivement. Si l'utilisateur n'entre pas une somme valide, le programme affiche un message d'erreur à cet effet, et se termine en affichant un message de fin. Lorsqu'une somme valide est saisie par l'utilisateur, le programme calcule les points gagnés comme suit :

1) Lorsque l'utilisateur entre une somme paire :

- Si la somme est strictement plus petite que 13, il gagne ($\text{somme} / 4$) points (où la division est une division entière).
- Si la somme est entre 13 et 22 inclusivement, il gagne ($\text{somme} / 2$) points (où la division est une division entière).
- Dans tous les autres cas, il gagne (somme) points.

2) Lorsque l'utilisateur entre une somme impaire, il gagne 1 seul point.

Dans tous les cas d'une somme valide (que la somme soit paire ou non), si la somme entrée est un multiple de 5, les points gagnés selon les conditions énumérées précédemment sont multipliés par 5. Finalement, le programme affiche les points gagnés et se termine en affichant un message de fin.

Attention, il n'y a aucune boucle dans cet exercice.

Note :

- On suppose que l'utilisateur saisit un nombre entier pour la somme (et pas un nombre réel ou un caractère ou une chaîne de caractères.).
- Utilisez l'opérateur modulo pour déterminer les multiples de 5.

Voici quelques exemples d'exécution de cet algorithme (s'il était implémenté dans un vrai langage de programmation) :

```
Ce programme calcule les points gagnés selon la somme de 5 des lances.
```

```
Entrez la somme des 5 des : 4
```

```
Erreur. Somme invalide.
```

```
FIN
```

```
Ce programme calcule les points gagnés selon la somme de 5 des lances.
```

```
Entrez la somme des 5 des : 5
```

```
Point(s) gagne(s) : 5
```

```
FIN
```

```
Ce programme calcule les points gagnes selon la somme de 5 des lances.  
Entrez la somme des 5 des : 12  
Point(s) gagne(s) : 3  
FIN
```

```
Ce programme calcule les points gagnes selon la somme de 5 des lances.  
Entrez la somme des 5 des : 20  
Point(s) gagne(s) : 50  
FIN
```

```
Ce programme calcule les points gagnes selon la somme de 5 des lances.  
Entrez la somme des 5 des : 22  
Point(s) gagne(s) : 11  
FIN
```

```
Ce programme calcule les points gagnes selon la somme de 5 des lances.  
Entrez la somme des 5 des : 25  
Point(s) gagne(s) : 5  
FIN
```

```
Ce programme calcule les points gagnes selon la somme de 5 des lances.  
Entrez la somme des 5 des : 30  
Point(s) gagne(s) : 150  
FIN
```

Exercice 2 (30 points)

Écrivez l'algorithme en pseudocode d'un programme qui calcule des statistiques sur une suite de nombres entiers saisis par l'utilisateur. Le programme débute en affichant le message de présentation suivant : *"Ce programme calcule des statistiques sur une suite d'entiers."* Ensuite, le programme demande à l'utilisateur d'entrer des nombres entiers strictement positifs (> 0), l'un à la suite de l'autre. L'utilisateur peut saisir autant de nombres qu'il le désire, et lorsqu'il veut terminer la saisie, il doit entrer le nombre 0. Lors de la saisie des entiers, si l'utilisateur entre un nombre plus petit que 0 (< 0), le programme affiche un message d'erreur, et redemande à l'utilisateur d'entrer un nombre entier. Lorsque l'utilisateur termine la saisie en entrant le nombre 0, le programme affiche les statistiques suivantes :

- Le nombre d'entiers valides (> 0) saisis par l'utilisateur.
- La somme des entiers valides (> 0) saisis par l'utilisateur.
- Le nombre de multiples de 5 saisis par l'utilisateur.

Et le programme se termine.

Attention, il n'y a qu'une seule boucle dans cet algorithme, et vous devez utiliser une boucle tant que.

Notes :

- On suppose que l'utilisateur entre bien des nombres entiers (et non des nombres réels ou des caractères, ou des chaînes de caractères...).
- Utilisez l'opérateur modulo pour déterminer les multiples de 5.

Voici deux exemples d'exécution de cet algorithme (s'il était implémenté dans un vrai langage de programmation) :

```
Ce programme calcule des statistiques sur une suite d'entiers.
```

```
Entrez un nombre entier > 0 (ou 0 pour terminer) : 0
```

```
STATISTIQUES
```

```
Nombre(s) saisi(s) : 0
```

```
Somme : 0
```

```
Multiple(s) de 5 : 0
```

```
Ce programme calcule des statistiques sur une suite d'entiers.
```

```
Entrez un nombre entier > 0 (ou 0 pour terminer) : 12
```

```
Entrez un nombre entier > 0 (ou 0 pour terminer) : 10
```

```
Entrez un nombre entier > 0 (ou 0 pour terminer) : 108
```

```
Entrez un nombre entier > 0 (ou 0 pour terminer) : -1
```

```
Erreur... Recommencez !
```

```
Entrez un nombre entier > 0 (ou 0 pour terminer) : -5
```

```
Erreur... Recommencez !
```

```
Entrez un nombre entier > 0 (ou 0 pour terminer) : 234
```

```
Entrez un nombre entier > 0 (ou 0 pour terminer) : 45
```

```
Entrez un nombre entier > 0 (ou 0 pour terminer) : 6
```

```
Entrez un nombre entier > 0 (ou 0 pour terminer) : 0
```

```
STATISTIQUES
```

```
Nombre(s) saisi(s) : 6
```

```
Somme : 415
```

```
Multiple(s) de 5 : 2
```

Exercice 3 (36 points)

Écrivez l'algorithme en pseudocode d'un programme qui calcule les taxes (TPS et TVQ) sur un montant donné. Le programme débute en affichant le message de présentation suivant : *"Ce programme calcule les taxes sur un montant donne."*

- Le programme demande ensuite à l'utilisateur d'entrer un montant (celui dont on veut calculer les taxes). Si le montant saisi est strictement plus petit que 0, le programme affiche un message d'erreur, sinon, il affiche le montant des deux taxes pour ce montant : TPS et TVQ .
- Ensuite, le programme demande à l'utilisateur s'il veut entrer un autre montant. Les seules réponses valides sont les caractères 'O' ou 'o' pour signifier OUI ou 'N' ou 'n' pour signifier NON. Si la réponse est OUI, le programme reprend à l'étape A). Si la réponse est NON, le programme se termine en

affichant un message de fin. Si la réponse est invalide (et tant que la réponse est invalide), le programme affiche un message d'erreur à cet effet, et redemande à l'utilisateur s'il veut entrer un autre montant (OUI ou NON).

Attention, cet algorithme contient DEUX boucles imbriquées. Utilisez les boucles de votre choix.

Notes :

- *On suppose que l'utilisateur entre bien un nombre réel pour le montant (et non un caractère, ou une chaîne de caractères), et qu'il entre bien un caractère pour la réponse oui / non (et non une chaîne de caractères).*
- *Lors de l'affichage de la TPS et de la TVQ, ne vous préoccupez pas d'arrondir les montants à deux décimales.*

Formules pour calculer la TPS et la TVQ :

Taux de TPS = 5 %

Taux de TVQ = 9.975 %

Montant TPS = Montant avant taxes * (Taux de TPS / 100)

Montant TVQ = Montant avant taxes * (Taux de TVQ / 100)

Voici deux exemples d'exécution de cet algorithme (s'il était implémenté dans un vrai langage de programmation) :

```
Ce programme calcule les taxes sur un montant donne.
```

```
Entrez un montant : -1
```

```
Erreur. Impossible de calculer les taxes sur un montant negatif.
```

```
Entrer un autre montant (O)ui ou (N)on ? g
```

```
Erreur. Vous devez entrer O ou N. Recommencez...
```

```
Entrer un autre montant (O)ui ou (N)on ? 1
```

```
Erreur. Vous devez entrer O ou N. Recommencez...
```

```
Entrer un autre montant (O)ui ou (N)on ? O
```

```
Entrez un montant : 34.56
```

```
TPS : 1.7280000000000002
```

```
TVQ : 3.44736
```

```
Entrer un autre montant (O)ui ou (N)on ? o
```

```
Entrez un montant : 19034
```

```
TPS : 951.7
```

```
TVQ : 1898.6415000000002
```

```
Entrer un autre montant (O)ui ou (N)on ? N
```

```
AUREVOIR
```

```
Ce programme calcule les taxes sur un montant donne.
```

```
Entrez un montant : 189.76
```

```
TPS : 9.488
```

```
TVQ : 18.92856
```

```
Entrer un autre montant (O)ui ou (N)on ? n
```

```
AUREVOIR
```

2. Date et modalité de remise

2.1 Remise

Date de remise : 2 octobre 2016 avant minuit (ou le 3 octobre À MINUIT).

Le fichier à remettre : TP1.txt (complété) – (PAS dans une archive zip, rar...)

Remise via Moodle uniquement.

Vous devez remettre (téléverser) votre fichier sur le site du cours (Moodle). Vous trouverez la boîte de remise « REMISE DU TP1 » dans la section **BOÎTES DE REMISE - TRAVAUX PRATIQUES**.

2.2 Politique concernant les retards

Une pénalité de 10% de la note finale, par jour de retard, sera appliquée aux travaux remis après la date limite. La formule suivante sera utilisée pour calculer la pénalité pour les retards :

$\text{Nbr points de pénalité} = m / 144$, où m est le nombre de minutes de retard par rapport à l'heure de remise. Ceci donne 10 points de pénalité pour 24 heures de retard, 1.25 point de pénalité pour 3 heures, etc.

Aucun travail ne sera accepté après 1 jour de retard et la note attribuée sera 0.

2.3 Remarques générales

- Aucun travail reçu par courriel ne sera accepté.
- Vous avez la responsabilité de **conserver des copies de sauvegarde** de votre travail (sur disque externe, Dropbox, Google Drive, etc.). La perte d'un travail due à un vol, un accident, un bris... n'est pas une raison valable pour obtenir une extension pour la remise de votre travail.
- N'oubliez pas d'écrire votre nom complet, et votre code permanent à l'endroit indiqué dans le fichier à remettre TP1.txt.
- N'attendez pas à la dernière minute, commencez tout de suite !

Le règlement sur le plagiat sera appliqué sans exception. Vous devez ainsi vous assurer de ne pas échanger du code avec des collègues ni de laisser sans surveillance votre travail au laboratoire. Vous devez également récupérer rapidement toutes vos impressions de programme au laboratoire.

BON TRAVAIL !