

## Énoncé 4

### Représentation et stockage de données Algorithmique et programmation avancées

élaboré par : Łukasz Starzak  
Department of Microelectronics and Computer Science, Lodz University of Technology

#### **Formatage / Salaire et prime**

1. Développez une fonction qui :

- prendrait un nombre d'heures de travail dans un mois, un code groupe d'employés (voir Tableau 1 dans l'énoncé 3) et un taux de prime (pourcentage) comme ses trois arguments ;
- obtiendra le salaire de base par heure correspondant en appelant la fonction développée dans l'exercice 2 de l'énoncé 3 (qui ne doit pas être modifiée ni son code copié vers la fonction d'ici ; cependant il faudra copier le fichier M correspondant vers le répertoire où se trouve le fichier M final créé dans le présent exercice) ;
- calculera le salaire mensuel de base (par une simple multiplication) ;
- obtiendra le salaire mensuel final correspondant en appelant la fonction développée dans l'exercice 4 de l'énoncé 2 (qui ne doit pas être modifiée ni son code copié vers la fonction d'ici ; cependant il faudra copier le fichier M correspondant vers le répertoire où se trouve le fichier M créé dans le présent exercice) ;
- ne renverra aucun résultat ;
- affichera les données d'entrée et celles calculées en alignant leurs valeurs à droite selon le modèle (prévoir assez de place pour accommoder des valeurs jusqu'à 99 999 €) :

Groupe employé	H
Salaire par heure	13.50 Eur
Temps de travail	125 h
Salaire de base	1687.50 Eur
Taux de prime	20.0 %
Salaire final	2025.00 Eur

#### **Fichiers / Liste de salaires**

2. \* Une entreprise a besoin de création automatisée de listes de salaires mensuels. Développez une fonction qui (voir les remarques à la fin de la description de cet exercice) :

a) lira les données d'entrée à partir d'un fichier texte (dont le nom est passé comme un premier argument) qui, dans chaque ligne, contient 3 colonnes :

- code du groupe d'employés,
- nombre d'heures de travail (nombre décimal),
- taux de prime (pourcentage entier),

et les placera dans une matrice ;

b) à partir de la matrice obtenue :

- obtiendra un vecteur de salaires de base par heure de la même manière que la fonction de l'exercice 1 (c'est-à-dire en appelant une fonction appropriée),
- calculera le salaire mensuel de base par une opération arithmétique,
- obtiendra le salaire mensuel final en appelant une fonction appropriée,
- calculera le montant de la prime (en euros) par une simple soustraction ;

et ajoutera tous ces résultats à la matrice d'entrée dans quatre colonnes supplémentaires ;

c) enregistrera les résultats obtenus (la matrice entière, y compris les données d'entrée) dans un nouveau fichier (dont le nom est passé comme un deuxième argument), où, dans chaque ligne, se trouveront, dans cet ordre :

- code du groupe d'employés (un caractère),
- nombre d'heures de travail (1 chiffre après le signe décimal),
- salaire de base par heure (2 chiffres après le signe décimal),
- salaire mensuel de base (2 chiffres après le signe décimal),
- taux de prime (sans partie décimale),
- montant de la prime (2 chiffres après le signe décimal),
- salaire mensuel final (2 chiffres après le signe décimal),

les valeurs dans chaque colonne étant alignées à droite ; les deux premières lignes de ce fichier doivent contenir les en-têtes des colonnes (p. ex. « Salaire par heure » etc.) ;

d) pour les colonnes du salaire mensuel de base, du montant de la prime et du salaire mensuel final (seulement celles-ci), calculera :

- leurs moyennes, en appelant la fonction de l'exercice 3b de l'énoncé 3,
- leurs valeurs maximales, en appelant la fonction de l'exercice 5a de l'énoncé 3,

et enregistrera ces résultats à la fin du fichier de sortie.

Des fichiers exemplaires d'entrée et de sortie sont disponibles sur la page web. Des différences de quelques centimes dans les sommes sont acceptables ; elles résulteront d'arrondis (on pourrait le remédier en intégrant les arrondis dans les fonctions de calcul développées mais ce n'est pas l'objectif de cet exercice).

### **Gestion d'erreurs / Code groupe d'employés**

3. Signalez une erreur si le code du groupe d'employé n'est pas correct.

a) Complétez la fonction développée dans l'exercice 2 de l'énoncé 3 de façon que, si le code passé à cette fonction n'est pas une lettre appartenant à l'ensemble prévu (voir Tableau 1 dans l'énoncé 3) :

- une erreur soit signalisée avec un communiqué explicatif,
- l'exécution du programme soit terminée.

Comparez l'effet de l'appel de la fonction de l'exercice 1 du présent énoncé : avec l'ancienne fonction « salaire groupe » et avec la fonction « salaire groupe » obtenue en ce sous-point, pour un cas où un code incorrect est passé.

b) \* Modifiez la fonction développée dans l'exercice 2 du présent énoncé :

- appliquez la structure *essaye-attrape* de façon que l'erreur provoquée par un code incorrect ne cause pas une terminaison immédiate du programme
- mais que le groupe des ouvriers soit supposé et les calculs continuent,
- toutefois en affichant un avertissement à l'utilisateur, précisant le numéro de la ligne du fichier d'entrée dans laquelle le code incorrect a été perçu.

### **Tri / Liste de salaires**

4. \* À la fonction de l'exercice 2 (ou 3b), ajoutez la fonctionnalité du tri des salariés avant l'enregistrement du fichier, selon les directives suivantes :

- pour réaliser le tri, créez une nouvelle fonction séparée qui triera une matrice passée comme son unique argument ;
- utilisez l'algorithme du tri à bulles (voir la conférence) ;
- la matrice doit être triée par rapport à la colonne du salaire mensuel final ;
- le tri doit être effectué dans l'ordre croissant ;
- notez que le tri doit être effectué par lignes complètes, ce qui veut dire qu'il faut intervertir des lignes entières et non pas seulement les deux éléments comparés.