

|  |
|--|
| <b><i>Le travail doit être remis <u>en version papier ET en version électronique</u></i></b> |
|--|

**Réalisation :** durant vos démonstrations et les périodes de Pratique Libre .

**Dépannage :** par vos démonstrateurs (lors des démonstrations), les surveillants des périodes de Pratique Libre

**Pénalité de retard :** Chaque jour de retard entraînera une pénalité de 5 points (sur 25 points) par jour.

**Note :** Le travail en équipe de deux (au maximum) est permis. Vous ne remettez alors qu'un travail par équipe avec les deux noms.

**Barème :** corrigé sur 25 points.

**Conseil amical :** N'attendez la semaine précédant la remise avant de commencer... vous n'aurez pas le temps!

---

**Matière exercée :** Éléments de base en langage C, structures et tableau des structures.

**Modalités de remise** (seront expliquées lors des séances de démonstration)

Assurez-vous d'avoir écrit vos coordonnées (nom et nom d'utilisateur) ainsi que ceux de votre coéquipier (s'il y a lieu) en commentaires au début de chaque programme. Les équipes de 2 personnes ne doivent remettre qu'une seule copie du T.P.

Les T.P. en retard (avec pénalité) peuvent aussi être déposés dans une boîte de remise.

## **La correction des travaux :**

Suite à une recommandation de la direction de la DESI et du DIRO, la qualité de l'algorithme et de la programmation occupent une place importante dans l'évaluation d'un travail.

**Un programme qui fonctionne est bien mais insuffisant pour mériter une meilleure note !**

Veuillez consulter les critères de correction de chacun des numéros d'un travail.

## Numéro 1 ( éléments de base du C : total de 10 points)

Le nombre 6 a 4 **diviseurs**. Ce sont 1, 2, 3 et 6.

Les **diviseurs propres** de 6 sont 1, 2 et 3 (le nombre 6 est exclu).

La somme des diviseurs propres de 6 sont  $1 + 2 + 3$  qui est égale aussi à 6. On dit que 6 est un **nombre parfait** (un nombre dont la valeur est égale à la somme de ses diviseurs propres).

Écrire un programme en langage C (suffixe .c) qui :

- 1) comporte **une fonction avec return**. Cette fonction permet de calculer et de retourner la somme des diviseurs propres d'un entier donné.
- 2) affiche la somme des diviseurs propres en utilisant la fonction écrite dans 1)
  - de 6
  - de 28
  - de 2004
- 3) comporte **une fonction de type void**. Cette fonction permet de calculer et de **transmettre via pointeur** la somme des diviseurs propres d'un entier donné.
- 4) affiche la somme des diviseurs propres en utilisant la fonction écrite dans 3)
  - de 6
  - de 28
  - de 2004
- 5) comporte une fonction permettant de retourner 1 (vrai) ou 0 (faux) selon qu'un entier donné est parfait ou non
- 6) permet de déterminer et d'afficher tous les nombres parfaits entre 5 et 10 000 avec leurs rangs selon le modèle suivant :

Les nombres parfaits entre 5 et 10000 :

- 1)  $6 = 1 + 2 + 3$
- 2)  $28 = 1 + 2 + 4 + 7 + 14$
- 2) etc ....

Rappels :

1) 14 est un diviseur de 28 car le reste dans la division de 28 par 14 vaut zéro :  $28 \% 14 == 0$

2) Le type booléen (boolean) n'existe pas en C. On utilise souvent le type entier (int) pour programmer le type booléen.

Exemple : niveau IFT 1810 (cours préalable)

```
int estAdulte ( int age )
{
    if (age >= 18)
        return 1 ; /* VRAI en C */
    else return 0 ; /* FAUX en C */
}
```

niveau IFT 1160 (notre cours) :

```
int estAdulte ( int age ) {
    return age >= 18;
}
```

### Critères de correction du numéro 1 (total de 10 points) :

- |   |            |
|---|------------|
| 1. Bon fonctionnement                         | : 3 points |
| 2. Fonctions et leurs utilisations            | : 2 points |
| 3. Qualité du codage                          | : 3 points |
| 4. Commentaires raisonnables                  | : 1 point  |
| 5. Présentation du programme et des résultats | : 1 point  |

Les données 6, 28, 2004 peuvent être déclarées comme constantes ou tapées (sans avoir besoin de valider) au clavier.

## Numéro 2 (Fichiers, tableaux de structures, recherche et fonctions: total de 15 points) :

### Description:

Soit le fichier suivant "**Employes.dat**" (copier les données de l'énoncé pour créer le fichier) contenant l'information au sujet d'un employé comme suit (c'est une structure) :

- une chaîne de 25 caractères représente le nom de l'employé,
- un seul caractère représente son poste de travail comme suit:  
    **A** pour analyste, **P** pour programmeur, **O** pour opérateur et **S** pour secrétaire
- Deux réels indiquant le nombre d'heures de travail par semaine et le taux horaire

Le fichier contient au maximum 30 lignes (30 employés).

Écrire un programme utilisant les tableaux et des fonctions qui permet de :

- lire le fichier, mémoriser les données dans un tableau de structures et compter le nombre effectif d'employés à traiter
- compter et d'afficher:
  - le nombre de programmeurs qui travaillent plus de 45 heures / semaine
  - le nombre d'opérateurs qui travaillent plus de 40.50 heures / semaine
  - le nombre de secrétaires qui travaillent plus de 35.00 heures / semaine
- afficher aussi :
  - les noms et les salaires hebdomadaires des opérateurs;
  - les noms et les salaires hebdomadaires des programmeurs
- lire le nom d'un employé, de le rechercher (utiliser obligatoirement la recherche dichotomique) et d'afficher son poste s'il est là ou un message s'il n'est pas là.

### Contraintes de l'interface :

- utiliser un interface avec menu présentant les différentes options à l'utilisateur.

### Le fichier **employes.dat**

|                   |   |      |       |
|-------------------|---|------|-------|
| Tremblay Alain    | A | 35.0 | 35.5  |
| Vachon Jean       | P | 40.0 | 22.75 |
| Lapalme Justin    | O | 40.0 | 15.75 |
| Deschenes Sylvie  | P | 35.0 | 25.0  |
| Lachance Carl     | O | 37.5 | 18.0  |
| Labonte Chantal   | P | 40.0 | 20.0  |
| Doucet Michel     | A | 40.0 | 33.75 |
| Desjardins Alex   | P | 35.0 | 25.0  |
| Tardif Guy        | A | 40.0 | 28.5  |
| Clinclin Stephane | O | 40.0 | 20.75 |
| Lafleur Marie     | A | 37.5 | 32.75 |
| Desbiens Robert   | P | 35.0 | 25.0  |
| Desautels Maryse  | P | 35.0 | 26.0  |
| St-germain guy    | O | 37.5 | 15.0  |
| Bourgeois Louis   | A | 37.5 | 29.0  |
| St-amour Flavie   | P | 40.0 | 25.0  |

#### Critères de correction du numéro 2 (total de 15 points) :

Pour chacune des fonctions demandées :

- |  |            |
|--|------------|
| 1. Écriture des fonctions                  | : 4 points |
| 2. Utilisation du fichier                  | : 3 points |
| 3. Utilisation des structures              | : 3 points |
| 3. Qualité du codage                       | : 3 points |
| 4. Commentaires, présentation et résultats | : 2 points |