

Durée : 1h30

Toutes les fonctions que vous écrirez doivent être clairement commentées avec des affirmations significatives (antécédents, conséquents, invariants). Vous prendrez soin de définir les bons paramètres et les bons types des données manipulées. Pensez à définir des fonctions auxiliaires si cela est nécessaire.

Soit la déclaration du type `Date` suivante :

► 1. Uniquement avec la notation de pointeur, écrivez la fonction :

qui renvoie le nom du mois correspondant à l'entier $m \in [1; 12]$.

[illegible]

Exercice 2

- 3. Avec la notation de pointeur (*i.e.* pas de tableau), écrivez la fonction `strrchr` qui renvoie un pointeur sur la dernière occurrence d'un caractère dans une chaîne de caractères. Le caractère et la chaîne de caractères sont passés en paramètre de la fonction. La fonction renvoie `NULL` si le caractère n'appartient pas à la chaîne. Attention : vous ne devez utiliser aucune fonction de la bibliothèque standard C.

This image shows a full page of blank, lined paper. It features approximately 20 evenly spaced horizontal grey lines across its entire width, providing a guide for handwriting or typing. The paper itself is a clean, off-white color.

Exercice 3

- 4. Écrivez la déclaration du type `liste` pour représenter une liste dynamique simplement chaînée, comme vue en TD.

- 5. Écrivez la procédure `supprimer` dont l'en-tête est donné ci-dessous :

```
/* Rôle : supprime l'élément de rang r de la liste l */
/* Antécédent :  $1 \leq r \leq \text{longueur}(l)$  */
void supprimer(liste *l, const int r)
```

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Exercice 4

- 6. Écrivez un programme qui prend une suite, non vide, de paramètres programmes qui doivent représenter des entiers. Le programme affichera sur la sortie standard la somme des entiers pairs. Vous vérifierez la validité des paramètres programmes. Vous pourrez utiliser, sans l'écrire, la fonction `estUnEntier` vue en TD. Votre programme ne devra utiliser que la notation de pointeur (pas de tableau).

Par exemple, si le programme exécutable se nomme `somEntiersPairs`, la commande suivante :

```
./somEntiersPairs 3 19 4 2 7 10
```

écrira sur la sortie standard : 16.

[illegible]

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.