

Langage C (V. Granet)

Durée : 1h30

Aucun document autorisé - Téléphones interdits

Nom chinois :

Nom Pinyin :

Numéro étudiant :

Classe :

- 1. En utilisant uniquement la notation de pointeur et à l'exclusion de toute autre fonction, écrivez en C la fonction `nEmeOccurrence` qui renvoie un pointeur sur la n^{e} occurrence d'un caractère dans une chaîne. Si la n^{e} occurrence n'existe pas, la fonction renverra `NULL`. Par exemple, l'appel `nEmeOccurrence("abcaeer", 'a', 2)`, renvoie l'adresse du 2ème a, et l'appel suivant : `nEmeOccurrence("abcaeer", 'a', 10)` renvoie `NULL`. Cette fonction possède l'en-tête suivant :

```
char *nIemeOccurrence(char *s, const char c, const int n)
```

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.

On souhaite représenter une liste d'éléments de type `T` de façon circulaire (le dernier élément est lié au premier) par une structure simplement chaînée. Pour cela, on définit le type `LISTE_CIR` suivant :

```
typedef int T; // par exemple
```

```
struct noeud {
    T elt;
    struct noeud *suivant;
};
```

```
typedef struct {
    struct noeud *first, *last;
} LISTE_CIR;
```

- 2. En utilisant cette déclaration de `LISTE_CIR`, dessinez une liste à 0 élément, une liste à 1 élément, une liste à 2 éléments, et enfin une liste à 3 éléments.

This image shows a single page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, leaving small margins at the top and bottom. There are no vertical margin lines, text, or other markings on the page.

- 3. Écrivez la procédure **tourner** qui prend en paramètre une liste **li** et un entier **n** et qui fait **n** fois le tour de la liste de façon circulaire.

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.

- 4. Écrivez la procédure `ajouterEnTete` qui ajoute un élément au début de la liste circulaire. Cette procédure possède l'en-tête suivant (que vous devez respecter) :

```
void ajouterEnTete(LISTE_CIR *li, T e)
```

[illegible]

- 5. Donnez la déclaration d'une structure `Date` pour représenter une date formée de 3 entiers : `jour`, `mois` et `année`.

- 6. Un fichier de caractères contient une suite d'entiers séparés par un ou plusieurs espaces. Trois entiers consécutifs représentent une date. Écrivez la fonction `creerFichDates` qui lit un fichier de caractères contenant la suite d'entiers, et qui crée un fichier de dates valides (de type `Date`). Sans l'écrire, vous pourriez utiliser la fonction booléenne `dateValide` qui teste si la date (de type `Date`) passée en paramètre correspond à une date valide ou pas. La fonction `creerFichDates` prend les noms des deux fichiers en paramètre. Son en-tête est le suivant :

```
void creerFichDates(char *in, char *out);
```

This image shows a single sheet of white paper with horizontal blue or grey 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.

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.