

Examen de Langage C (V. Granet)

Durée : 1h30
Aucun document autorisé
Mobiles interdits

- 1. À l'aide de la notation de pointeur, écrivez en C la fonction `strdup` qui renvoie la duplication de la chaîne de caractères passée en paramètre. Cette fonction possède l'en-tête suivant, et que vous devez respecter : `char *strdup(const char *s)`;

```
/*
 * Rôle : renvoie la duplication de s
 */
char *strdup(const char *s) {
    // allouer l'espace pour tous les caractères de s
    char *s1 = malloc(strlen(s) + 1);
    return s1 ? strcpy(s1, s) : NULL;
}
```

- 2. Une liste d'entiers est représentée par une structure *simplement* chaînée et par la déclaration suivante :

```
typedef struct noeud {
    int elt;
    struct noeud *suivant;
} * Liste;
```

Écrivez en C la fonction `ajouterEnFin` qui ajoute un entier à la fin d'une liste d'entiers. La liste et l'entier sont passés en paramètres. L'en-tête de cette fonction, que vous devez respecter, est le suivant :

```
void ajouterEnFin(Liste *le, int x)
```

```
const Liste listeVide = NULL;
```

```
/*
 * Rôle : ajoute à la fin de liste *le l'entier x
 */
void ajouterEnFin(Liste *le, int x) {
    // créer le nouvel élément
    struct noeud *p = malloc(sizeof(struct noeud));
    p->elt = x;
    p->suivant = NULL;
    // l'ajouter en fin de liste
    if (*le == listeVide) *le = p;
    else {
        // la liste contient au moins 1 élément =>
        // se placer sur le dernier élément
        struct noeud *q = *le;
        while (q->suivant != NULL)
            q = q->suivant;
    }
}
```

```
    // ajouter le nouvel élément à la fin
    q->suivant = p;
}
}
```

- 3. Un fichier contient une suite, éventuellement vide, de trames. Une trame est formée d'un entier (`int`) dont la valeur n indique le nombre d'octets qui forment le reste de la trame. En octets, quelle est la taille d'une trame ?

`sizeof(int)+n`

- 4. Écrivez en C le programme `trames.c` qui prend sur la ligne de commandes un nom de fichier et qui affiche le nombre de trames nulles qu'il contient. Une trame nulle est une trame dont les n (> 0) octets sont égaux à 0. *Vous ferez les vérifications de validité nécessaires*. Toutefois, on considère que les trames sont correctement formées.

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
/*
 * Rôle : renvoie le nombre de trames nulles contenues
 * dans le fichier s
 */
int compterTramesNulles(char *s) {
    // ouvrir le fichier en lecture
    FILE *fd;
    if ((fd=fopen(s, "r"))==NULL) {
        perror(s);
        exit(errno);
    }
    // ok, le fichier est bien ouvert en lecture
    int NbTramesNulles=0, n=0;
    while (fread(&n, sizeof(int), 1, fd)>0)
        if (n>0) {
            // lire n octets et vérifier s'ils sont égaux à 0
            // Note : il faut lire toute la trame
            int octetNul = 1;
            for (int i=0; i<n; i++) {
                char o;
                fread(&o, sizeof(char), 1, fd);
                if (octetNul && o!=0)
                    // un octet non nul
                    octetNul=0;
            }
            if (octetNul) NbTramesNulles++;
        }
    // fin de fichier
    fclose(fd);
    return NbTramesNulles;
}

int main(int argc, char *argv[]) {
    if (argc!=2) {
        fprintf(stderr, "Usage : %s file\n", argv[0]);
        exit(EXIT_FAILURE);
    }
}
```

```
}
// afficher le nombre de trames nulles
printf("nb trames nulles = %d\n", compterTramesNulles(argv[1]));
return EXIT_SUCCESS;
}
.....
```