

Examen de Examen de Langage C

Durée : 1h30

Aucun document autorisé – Mobiles interdits

Notez que les affirmations (antécédents, conséquents, rôles, et invariants) dans vos codes C entreront pour partie dans la note finale.

- 1. À l'aide d'un **typedef**, écrivez la déclaration du type **liste** pour représenter une liste linéaire dynamique simplement chaînée (comme vue en cours) de réels **double**.

```
typedef struct noeud {  
    double elt;  
    struct noeud *suivant;  
} *liste;
```

- 2. Écrivez la procédure **ajouterEnTete** qui ajoute dans une liste (du type précédemment déclaré) un réel **double** en début de liste. Cette procédure a deux paramètres.

```
/*  
 * Rôle : ajoute x en tête de la liste l  
 */  
void ajouterEnTete(liste *l, const double x) {  
    struct noeud *p = malloc(sizeof(struct noeud));  
    p->elt = x;  
    p->suivant = *l;  
    *l = p;  
}
```

- 3. Soit *f* un fichier qui contient une suite de réels **double** (attention, ce n'est pas un fichier de texte!). Écrivez la procédure **creerListeDeDouble** qui lit tous les réels du fichier *f* et les place dans une liste (du type précédemment déclaré). Cette procédure possède deux paramètres la liste et le nom du fichier (une chaîne de caractères). Évidemment, vous pourrez utiliser la procédure **ajouterEnTete**.

```
/*  
 * Rôle : crée la liste de double l à partir des doubles  
 *        contenu dans le fichier f  
 */  
void creerListeDeDouble(liste *l, const char *f) {  
    FILE *fp;  
    if ((fp = fopen(f, "r"))==NULL) {  
        perror(f);
```

```
        exit(errno);  
    }  
    // le fichier de double est ouvert en lecture  
    initListe(l);  
    double x;  
    while (fread(&x, sizeof(double), 1, fp)>0)  
        ajouterEnTete(l, x);  
    //  
    fclose(fp);  
}
```

- 4. Écrivez la procédure **creerFichierDoublesPositifs** qui, à partir d'une liste de réels **double**, fabrique un fichier qui ne contient que des réels positifs (> 0.) Cette procédure possède deux paramètres la liste et le nom du fichier (une chaîne de caractères).

```
/*  
 * Rôle : crée le fichier de double à partir des réels positifs  
 *        contenus dans l  
 */  
void creerFichierDoublesPositifs(const liste l, const char *f) {  
    FILE *fp;  
    if ((fp = fopen(f, "w"))==NULL) {  
        perror(f);  
        exit(errno);  
    }  
    // le fichier de double est ouvert en écriture  
    struct noeud *p = l;  
    while (p!=NULL) {  
        double x = p->elt;  
        if (x>0)  
            // écrire le réel positif dans le fichier f  
            fwrite(&x, sizeof(double), 1, fp);  
        //  
        p = p->suivant;  
    }  
    fclose(fp);  
}
```

- 5. On veut un programme qui prend comme paramètres programmes deux noms de fichiers. Le premier est le nom d'un fichier qui contient une suite (éventuellement vide) de réels **double** (attention, ce n'est pas un fichier de texte!). Le second est le nom du fichier qui contiendra les réels positifs issus du premier fichier.

Dans la fonction **main**, écrivez le code nécessaire pour créer le second fichier de réels positifs. Vous ferez toutes les vérifications nécessaires.

```
#define USAGE "myprog f1 f2"
```

```
/*  
 * Rôle : écrit le message d'erreur msg sur la sortie d'erreur std
```

```

*           et arrête le programme avec le code de retour EXIT_FAILURE
*/
void erreur(char *msg) {
    fprintf(stderr, "%s\n", msg);
    exit(EXIT_FAILURE);
}

int main(int argc, char *argv[]) {
    // Vérifier le nombre de paramètres programmes
    if (argc!=3) erreur(USAGE);

    liste l;
    creerListeDeDouble(&l, argv[1]);
    creerFichierDoublesPositifs(l, argv[2]);

    return EXIT_SUCCESS;
}

.....

```