

Examen de Langage C (V. Granet)

Durée : 1h30

Aucun document autorisé

Mobiles interdits

- 1. Écrivez la structure `compteurs` qui contient deux champs de type `int` appelés `nbLettres` et `nbChiffres`.

```
struct compteurs {
    int nbLettres;
    int nbChiffres;
};
```

- 2. En utilisant uniquement la notation de pointeur, écrivez la fonction `compter` qui renvoie une structure de type `compteurs` qui contient le nombre de lettres et de chiffres contenus dans la chaîne de caractères passée en paramètres. Vous pourrez utiliser les fonctions `isalpha` et `isdigit`.

```
/*
 * Rôle : renvoie le nombre de lettres et de chiffres contenus
 *        dans la chaîne de caractères s
 */
struct compteurs compter(const char *s) {
    struct compteurs res = {0,0};
    while (*s) {
        if (isalpha(*s)) res.nbLettres++;
        else
            if (isdigit(*s)) res.nbChiffres++;
        s++;
    }
    return res;
}
```

- 3. En utilisant la fonction `compter` précédente, écrivez en C le programme `myprog` qui affiche sur la sortie standard le nombre de lettres et de chiffres d'une chaîne de caractères passée en paramètre au programme par l'intermédiaire de `argc` et `argv`. Vous ferez les vérifications nécessaires des paramètres. Par exemple, les exécutions du programme `myprog` suivantes donneront :

```
./myprog toto123+-xx9
6 4
./myprog
Error! Usage : myprog string
```

```
int main(int argc, char *argv[]) {
    if (argc!=2) {
        fprintf(stderr, "Usage : compter str\n");
        return EXIT_FAILURE;
    }
}
```

```
}
struct compteurs r = compter(argv[1]);
printf("%d %d\n", r.nbLettres, r.nbChiffres);
return EXIT_SUCCESS;
}
```

- 4. Un fichier contient une suite de réels **double** (attention ce n'est pas un fichier de caractères). À partir de ce fichier, on veut fabriquer une liste chaînée de **double**. La liste est définie par la déclaration de type suivante :

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

Écrivez la procédure `creerListeD` qui crée une liste de réels **double** à partir d'un fichier de **double**. Le nom du fichier est passé en paramètre. Vous vérifierez la validité de l'ouverture du fichier. Pensez à insérer chaque nouveau réel **double** lu en tête de liste. L'en-tête de cette procédure (que vous devez respecter) est le suivant :

```
/*
 * Rôle : crée la liste de double l à partir du fichier
 *        de double de nom f
 */
void creerListeD(const char *f, listeD *l) {

}

/*
 * Rôle : insère en tête de liste l le réel double x
 */
void insererEnTete(const double x, listeD *l) {
    struct noeud *n = malloc(sizeof(struct noeud));
    n->elt = x;
    n->suivant = *l;
    *l = n;
}

/*
 * Rôle : crée la liste de double l à partir du fichier
 *        de double de nom f
 */
void creerListeD(const char *f, listeD *l) {
    FILE *fd = fopen(f, "r");
    double x;

    *l=NULL;
    while (fread(&x, sizeof(double), 1, fd)>0)
        insererEnTete(x, l);
    fclose(fd);
}
```

- 5. Écrivez la procédure `printListeD` qui affiche sur la sortie standard tous les réels **double** de la liste. Cette procédure possède l'en-tête suivant :

```
/*
```

```

* Rôle : écrit sur la sortie standard les réels double
*      contenus dans la liste l
*/
void printListeD(listeD l) {

```

```

/*
* Rôle : écrit sur la sortie standard les réels double
*      contenus dans la liste l
*/
void printListeD(listeD l) {
    while (l!=NULL) {
        printf("%f ", l->elt);
        l=l->suivant;
    }
    printf("\n");
}

```

```

.....

```