

Examen de Langage C

Durée : 1h30

Aucun document autorisé

Mobiles interdits Documents non autorisés

Note : la qualité des commentaires, avec notamment la présence d'affirmations significatives et d'invariant, ainsi que les noms donnés aux variables, et la bonne indentation rentreront pour une part importante dans l'appréciation du travail.

- 1. En spécifiant des affirmations pertinentes, vous expliquerez le principe de son algorithme et ce que calcule la fonction `mystere` suivante :

```
int mystere(int a, int b) {
    int x=0, y=a;
    while (y>=b) {
        int k=1, p=b, z=b<<1;
        while (z<=y) {
            p=z;
            z<<=1;
            k<<=1;
        }
        y-=p;
        x+=k;
    }
    return x;
}
```

Question sur 3 pts : Cette fonction calcule et renvoie le quotient de la division entière de a par b (avec $a \geq 0$ et $b > 0$). Elle procède comme l'algorithme, vu en TD, par soustractions successives, mais l'améliore de façon significative. Au lieu de retirer, à chaque itération, 1 diviseur au dividende et d'incrémenter de 1 le quotient, le diviseur est multiplié successivement par 2 pour trouver son plus grand multiple que l'on peut soustraire au dividende. Le quotient sera alors incrémenté de ce nombre de multiples. Après soustraction, on recommence ce processus sur le reste obtenu qui devient le nouveau dividende. Les multiplications par 2 sont réalisées par des décalages binaires et améliorent encore l'efficacité de l'algorithme.

Corollaire : vous pouvez constater que l'emploi de noms de variables non significatifs, et l'absence d'affirmation rend difficile la compréhension des algorithmes et des programmes.

- 2. Donnez le résultat de l'appel suivant : `mystere(2345,23)`.

Question sur 1 pt : `mystere(2345,23)=101`

-
- 3. Écrivez en C, un programme qui lit toute l'entrée standard, et qui compte et affiche le nombre (différent de 0) de chacune des 26 lettres minuscules qu'il contient. On affichera aussi le nombre total de caractères lus, ainsi que le nombre de caractères qui ne sont pas des lettres minuscules. Par exemple, si l'entrée standard contient les 2 lignes suivantes :

ces exercices de programmation C,
quel pensum !

votre programme devra afficher sur la sortie standard :

l'entrée standard comporte 47 caractères dont :

```
2 fois la lettre a
3 fois la lettre c
1 fois la lettre d
7 fois la lettre e
1 fois la lettre g
2 fois la lettre i
1 fois la lettre l
3 fois la lettre m
2 fois la lettre n
2 fois la lettre o
2 fois la lettre p
1 fois la lettre q
3 fois la lettre r
3 fois la lettre s
1 fois la lettre t
2 fois la lettre u
1 fois la lettre x
et 10 autres caractères
```

Question sur 6 pts :

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
```

```
#define NBLETTRES 26
```

```
int main(void) {
    int c, autres, nbcars, lettres[NBLETTRES] = {0};

    autres = nbcars = 0;
    while ((c=getchar())!=EOF) {
        nbcars++;
        if (islower(c))
            // incrémenter le nombre de c minuscule
            lettres[c-'a']++;
        else
            autres++;
    }
    // afficher les occurrences, différentes de 0,
    // de chaque lettre minuscule
    printf("l'entrée standard comporte %d caractères dont :\n", nbcars);
```

```

for (int c=0; c<NBLETTRES; c++)
    if (lettres[c]!=0)
        printf("%3d fois la lettre %c\n", lettres[c], c+'a');
// afficher le nombre de caractères « autres »
if (autres!=0)
    printf("et %d autres caractères\n", autres);
return EXIT_SUCCESS;
}

```

4. Soit une matrice $M \times N$ d'entiers (`int`) (avec $M > 0$ et $N > 0$), écrivez en C la fonction `sommeContour` qui renvoie la somme des valeurs d'une sous-matrice $m \times n$ ($0 < m \leq M$ et $0 < n \leq N$) qui sont les plus à l'extérieur de la matrice (celles de la première et dernière ligne, et de la première et dernière colonne). Par exemple, pour la sous-matrice 3×4 suivante, c'est la somme des entiers en

rouge : $\begin{vmatrix} 2 & 3 & 1 & -4 \\ 1 & 6 & 9 & -2 \\ -5 & 7 & 1 & 0 \end{vmatrix}$ qui vaut 4. L'en-tête de la fonction est le suivant :

```
int sommeContour(int mat[][N], int m, int n)
```

Question sur 4 pts : dans cet exercice, il s'agit évidemment de ne parcourir que le contour de la sous-matrice, et de factoriser les parcours des lignes et des colonnes !

```

/*
 * Antécédent : 0 < m ≤ M et 0 < n ≤ N
 * Rôle : renvoie la somme des valeurs de contour de
 * la sous-matrice m×n
 */
int sommeContour(int mat[][N], int m, int n) {
    int i, s = 0;
    assert(m>0 && m≤M && n>0 && n≤N);
    /* parcourir la 1ère et la dernière ligne */
    for (i=0; i<n; i++)
        s+=mat[0][i]+mat[m-1][i];
    /* parcourir la 1ère et la dernière colonne */
    for (i=1; i<m-1; i++)
        s+=mat[i][0]+mat[i][n-1];
    return s;
}

```

5. Un palindrome est un mot ou un groupe de mots (séparés par des espaces) qui peut être lu indifféremment de gauche à droite ou de droite à gauche en conservant le même sens. radar et elu par cette crapule sont deux exemples de palindrome (on ne tient pas compte des espaces). Écrivez en C la fonction palindrome qui teste si une chaîne de caractères passée en paramètre est un palindrome.

Question sur 6 pts :

```

#include <string.h>
#include <ctype.h>

```

```

#include <stdbool.h>

/*
 * Rôle : vérifie si le mot ou groupe de mots (séparés par des
 * espaces) désigné par p est un palindrome ou non
 */
bool palindrome(const char p[]) {
    int j = strlen(p)-1;
    if (j<0) /* la phrase est vide */ return false;
    else {
        int i=0;
        bool estUnPalindrome=true;
        /* parcourir la phrase par les deux extrémités
         * sauter les espaces si nécessaire
         */
        while (i<j && estUnPalindrome) {
            if (isspace(p[i])) i++;
            else if (isspace(p[j])) j--;
            else if (p[i] != p[j]) estUnPalindrome=false;
            i++;
            j--;
        }
        return estUnPalindrome;
    }
}

```