Sujet 1 (29/11/2024)

contact : jean-baptiste.doderlein@ens-rennes.fr

Question de cours

1. Décrire le rôle de la pile et du tas dans la mémoire virtuelle d'un programme. Pour chacun d'entre eux, donner un exemple illustré (par un schéma et/ou du code C) de leur utilisation.

2. Quelle taille fait le tableau `chaine`? La chaine de caractère est-elle modifiable ?

```c
char* chaine = "Hello World!";
```

Exercice 1 : Algorithme d'Euclide

On donne la fonction suivante qui calcule le PGCD de deux entiers positifs :

```c
int pgcd(int a, int b){
    int x = a;
    int y = b;
    while(y!=0){
        if (x>y)
            x = x-y;
        else
            y = y-x;
    }
    return x;
}
```

1. Montrer que cette fonction termine.

2. Montrer que cette fonction calcule bien le PGCD

Indication : $pgcd(a, b) = pgcd(a - b, b)$ pour $a > b$ et a, b positifs

Exercice 2 : Matrices

On souhaite écrire des fonctions pour faire du calcul matriciel en C. On supposera dans un premier temps que les matrices sont carrées.

1. Écrire une fonction qui renvoie une matrice de taille n remplie de 0.

2. Écrire une fonction qui calcule la somme de deux matrices ($S[i][j] = A[i][j] + B[i][j]$)

3. Écrire une fonction qui calcule le produit de deux matrices ($P[i][j] = \Sigma_{k=0}^{N-1} A[i][k] \times B[k][j]$)

4. Écrire une fonction qui effectue une rotation d'une matrice.

Exemple de rotation :

```
1 2   ->  3 1
3 4       4 2
```

5. Écrire une rotation de matrice sans utiliser `malloc`.

Sujet 2 (29/11/2024)

contact : jean-baptiste.doderlein@ens-rennes.fr

Exercice de cours : Recherche dichotomique

L'algorithme de recherche dichotomique est un algorithme qui détermine si un élément est dans un tableau trié. On suppose que le tableau est trié par ordre croissant.

1. Écrire en C, sans utiliser de récursion, la fonction `bool recherche_dichotomique(int* t, size_t n, int x)`.

2. Montrer que cette fonction termine.

Exercice 2 : Des pointeurs et des erreurs

1. Donner la valeur de `x`, `y` et `z` pour chaque ligne de la fonction `main`

```c
void f1 (int a, int *b ) { a = *b; }
void f2 (int *b, int c) { *b = c; }
void f3 (int **b, int a) { **b = a; }
void f4 (int *a, int c) { f3(&a, c); }
void f5 (int **a, int *b, int c) { *a = b; **a = c; }
int main () {
    int x = 5, y = 7, z = 9;
    f1(x, &y);
    f2(&x, y);
    f4(&y, z);
    int* p = &y;
    f5(&p, &z, x);
}
```

2. Que fait ce code ? Est-ce qu'il compile ? Donner les différentes erreurs présentes dans ce code et proposez une version corrigée du code.

```c
int *mk_data() {
    int array[10] = {0};
    for (size_t i = 0; i <= 10; i++) {
        array[i] = i*i;
    }
    return array;
}
int main() {
    int *data = mk_data();
    for (size_t i = 0; i <= 10; i++) {
        printf("%i\n", data[i]);
    }
    free(data);
}
```

Exercice 3 : Fonction sur chaines de caractères

1. Écrire une fonction en C qui vérifie si une chaîne de caractères est un palindrome.

2. Écrire une fonction en C qui inverse une chaine de caractère en place.

3. Écrire une fonction qui réalise la concaténation de deux chaines de caractères. On pourra supposer que les chaines de caractères sont alloués sur le tas.

Sujet 3 (29/11/2024)

contact : jean-baptiste.doderlein@ens-rennes.fr

Question de cours

1. Donner la définition d'un variant de boucle.

2. Justifier en quoi la présence de variants de boucle dans un programme itératif permet de montrer sa terminaison.

Exercice 1 : Terminaison de la fonction 91

La fonction de McCarthy 91 est une fonction récursive définie comme suit :

$$f(n) = \begin{cases} n - 10 & \text{si } n > 100 \\ f(f(n+11)) & \text{si } n \leq 100 \end{cases}$$

1. Implémenter cette fonction en C.

2. Donner le résultat de la fonction pour $n = 110$ et $n = 98$.

3. Que peut on dire de la terminaison de la fonction pour $n > 100$? Montrer que la fonction termine pour tout n.

Exercice 2 : Implémentation d'une file en C

Vous devez implémenter une file d'entiers de taille bornée (par exemple, 10) en utilisant un tableau. Voici la structure de base de la file en C :

```c
#define TAILLE_MAX 10

typedef struct {
    int elements[TAILLE_MAX];
    int debut;
    int fin;
    size_t taille;
} File;
```

1. On considère une file F de taille 4. Illustrer la file quand : on ajoute les éléments 1,2 et 3, on retire un élément puis on ajoute les éléments 4 et 5.

2. Implémenter une fonction `bool estPleine(File *f)` qui vérifie si la file est pleine.

3. Implémenter une fonction `bool enfiler(File *f, int valeur)` qui ajoute un élément à la fin de la file si elle n'est pas pleine.

4. Implémenter une fonction `bool defiler(File *f, int *res)` qui retire un élément au début de la file si elle n'est pas vide.

5. Proposer une fonction pour tester la file.

6. Quel changement faudrait-il faire pour avoir une file de taille non bornée ?