

Sujet 1 (29/11/2024)

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

Question de cours

1. Expliquer les différents paramètres de : `gcc -O0 -Wall -Wextra -Werror -fsanitize=address,undefined fichier.c -o fichier`
2. Quelle taille fait le tableau `chaine`? La chaîne de caractère est-elle modifiable ?

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

Exercice 1 : Algorithme d'Euclide

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

```
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 : Javanais

Le javanais est un jeu linguistique où l'on insère "av" avant chaque voyelle ou groupe de voyelles. Par exemple, "un bon abricot" devient "avun bavon avabravicavot".

1. Écrire une fonction `javanais` qui insère "av" avant chaque voyelle du mot donné. Le prototype est :

```
void javanais(char* mot, char* buffer);
```

Exemple : "jour" → "javoavur"

2. Écrire la fonction `javanais2` pour insérer "av" avant un **groupe de voyelles consécutives**.

Exemple :

- "jour" → "javour"
- "eau" → "aveau"

Sujet 2 (29/11/2024)

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

Question de cours

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.

Exercice 1 : 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 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`

```
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); }
int main () {
    int x = 5, y = 7, z = 9;
    f1(x, &y);
    f2(&x, y);
    f3(&y, z);
    return 0;
}
```

2. Que fait ce code ? Est-ce qu'il compile ? Donner les différentes erreurs présentes dans ce code.

```
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);
    return 0;
}
```

Exercice 3 : Fonction sur chaînes 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 chaîne de caractère en place.

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 : 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. Montrer que cette fonction termine pour toute valeur d'entrée n à l'aide d'un variant de boucle.
3. Montrer que cette fonction renvoie toujours 91.

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 :

```
#define TAILLE_MAX 10

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

1. Implémenter une fonction `bool estVide(File *f)` qui vérifie si la file est vide.
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.