

Algorithmique et Structures de données 1

L2 2021-2022
Travaux Pratiques 2

Site du cours : <https://defelice.up8.site/algo-struct.html>

Les exercices marqués de (@) sont à faire dans un second temps.

Un fichier écrit en langage C se termine conventionnellement par `.c`.

Une commande de compilation est `gcc fichier_source1.c fichier_source2.c fichier_source3.c`.

Voici des options de cette commande.

- `-o nom_sortie` pour donner un nom au fichier de sortie (par défaut `a.out`).
- `-Wall` pour demander au compilateur d'afficher plus de Warnings
- `-Wextra` pour demander au compilateur d'afficher plus de Warnings
- `-std=c11` pour compiler selon la norme C11

Exemple : `gcc -Wall fichier1.c -o monprogramme`

Exercice 1. *Savez-vous compter ?*

Recopier et exécuter la fonction suivante avec `rebour(10)`. Elle compte à rebours.

```
void rebour(int n)
{
    if(n<0) return;
    printf("%d\n",n);
    rebour(n-1);
}
```

Sans modifier les trois instructions, mais simplement en modifiant leur ordre, faire compter la fonction à l'endroit.

Exercice 2. *Remplir récursivement*

Écrire une fonction `void remplirRecTab(int taille,int* tabl)` qui remplit le tableau `tabl` de taille `taille` avec des nombres aléatoires compris entre -100 et 100 (avec `rand` par exemple) sans utiliser de boucle (`for` ou `while`). (On suppose que la place mémoire du tableau doit déjà être allouée)

Exercice 3. *Somme de chiffres*

Écrire une fonction récursive `int sommeChiffres(int a)` qui renvoie la somme des chiffres provenant de l'écriture décimale du nombre `a`.

Exemple : $456 \rightarrow 4 + 5 + 6 = 15$

La fonction ne doit pas utiliser de boucle. Aide :

1. En C la division d'un nombre entier positif par un autre (positif aussi) `a/b` renvoie la partie entière inférieure du résultat réel. Exemple : L'expression `17/10` vaut 1 en C.
2. Vous pouvez utiliser l'opérateur `%`, `a % b` renvoie le reste de la division euclidienne de `a` par `b`. Exemple : `456%10` vaut 6 en C

Exercice 4. Recherche

Écrire une fonction `int rechercherRecTab(int taille,int* tab,int val)` qui renvoie l'indice de la case du tableau `tab` contenant la valeur `val`. Si le tableau ne contient pas la valeur, la fonction doit renvoyer `-1`. La fonction `rechercheRecTab` ne doit pas utiliser de boucle.

Exercice 5. Racine carre

Écrire une fonction `float racineCarree(float a,float erreur)` qui renvoie `s` une valeur approchée de la racine carrée de `a`. On doit avoir $|s^2 - a| < \text{erreur}$

Exercice 6. Tri simple

Écrire une fonction `void trierTab(int taille,int* tab)` qui permute les valeurs du tableau de façon à obtenir un tableau trié.

Exercice 7. Le sac à dos

Un campeur choisit avec soin les objets qu'il va emporter dans son sac à dos. Il doit choisir parmi une liste d'objets et chaque objet est associé à deux nombres entiers : sa valeur et sa masse. Le campeur ne doit pas dépasser une masse `m` tout en maximisant la somme des valeurs des objets.

Écrire une fonction `int sacADos(int tailledeMetV,int* M,int* V,int m)` qui donne la valeur maximum que l'on puisse obtenir au problème du sac à dos. Les tableaux `M` et `V` ont la même taille et si `i` est un entier la case `M[i]` est la masse de l'objet `i` et `V[i]` est sa valeur.