

Fonctions

1. Spécifications de fonctions

1. a. Ecrivez les spécifications de la fonction suivante :

```
void imprimer (const char *s , char **t) {
    if (t == NULL){
        printf("table absente\n");
        return;
    }
    if (*t == NULL){
        printf("table vide\n");
        return;
    }
    printf("%s\n", s);
    for (char **p = t; *p != NULL; p++) {
        printf("%s\n", *p);
    }
}
```

1. b. Ecrivez les spécifications de la fonction suivante :

```
void imprimer (const char *s , int *t, int taille) {
    if (t == NULL){
        printf("table absente\n");
        return;
    }
    if (taille == 0){
        printf("table vide\n");
        return;
    }
    printf("%s\n", s);
    for (int i=0; i<taille; i++) {
        printf("%i\n", t[i]);
    }
}
```

1. c. Ecrivez les spécifications de la fonction suivante :

```
int rechercher (const char *s, char c) {
    if (s == NULL){
        return -1;
    }
    for (char *p = s; *p != '\0'; p++) {
        if (*p == c)
            return p-s;
    }
    return -1;
}
```

2. Prototypes de fonctions

2. a. Ecrivez le prototype d'une fonction qui reçoit en argument le chemin complet d'un fichier. Elle renvoie vrai si un tableau passé en paramètre a été rempli avec les lignes du fichier converties en entiers et ses tailles physique et logique mises à jour ; faux sinon.
Nommez les paramètres de façon explicite.
2. b. Ecrivez le prototype d'une fonction qui reçoit en argument le chemin complet d'un fichier. Elle renvoie vrai si un tableau passé en paramètre a été rempli avec les mots du fichier et ses tailles physique et logique mises à jour ; faux sinon.
Nommez les paramètres de façon explicite.
2. c. Ecrivez le prototype d'une fonction qui sépare les mots d'une chaîne de caractères. Elle les sauve dans un tableau passé en paramètre, sauve sa taille physique fournie en paramètre et renvoie sa taille logique. Elle renvoie -1 si une erreur s'est produite.
Nommez les paramètres de façon explicite.

3. Implémentation de fonctions

3. a. Ecrivez une fonction qui permute la valeur de deux entiers.
3. b. Ecrivez une fonction qui permute le contenu de deux chaînes de caractères.
3. c. Ecrivez une fonction qui permute le contenu de deux tableaux d'entiers.

4. Débogage de fonction

Expliquez le(s) problème(s) que présente le code suivant et proposez une solution, sans modifier le prototype de la fonction :

```
/* Crée un tableau avec les puissances d'un nombre
 * PRE: n: nombre entier strictement positif
 * RES: renvoie un tableau de taille n+1
 *      rempli avec les puissances de n
 */
int* creerTable (int n) {
    int t[n+1];
    t[0] = 1;
    for (int i=1; i<=n; i++)
        t[i] = t[i-1] * n;
    return (int*)t;
}
```

5. Exercice récapitulatif

Ecrire un programme qui reçoit des arguments sur la ligne de commande et les traite comme suit :

- il copie les chaînes passées en argument dans un nouveau tableau de chaînes ;
- un argument ne sera copié dans cette table que s'il ne s'y trouve pas déjà (pas de doublons possibles) ;
- le programme trie la table par ordre alphabétique grâce à un tri par sélection, dont voici le pseudo-code :

```
procédure tri_selection (tableau t)
    n ← longueur(t)
    pour i de 0 à n-2
        min ← i
        pour j de i+1 à n-1
            si t[j] < t[min], alors min ← j
        fin pour
        si min ≠ i, alors échanger t[i] et t[min]
    fin pour
```

Nous vous demandons de compléter le fichier **ex7.5.c**, i.e. la fonction principale `main()` et les fonctions suivantes (les specs sont précisées dans le fichier source) :

```
char** copierArgs (char** tab, int n, int* ncp);
void trier (char** tab, int n);
void afficher (char** tab, int n);
```

Pour rappel, les principales fonctions traitant les chaînes sont : `strlen`, `strcpy`, `strcat` et `strcmp`. Référez-vous au `man` pour plus de détails sur leur utilisation.

Exemple d'exécution :

```
> ./a.out toton titi toto tata toto titi
Arguments non triés:
    1 - 'toton'
    2 - 'titi'
    3 - 'toto'
    4 - 'tata'
Arguments triés:
    1 - 'tata'
    2 - 'titi'
    3 - 'toto'
    4 - 'toton'
```