



Algorithmes et structures de données

TP 01

Gulfem Isiklar Alptekin – Ozgun Pinarer

Ecrivez les fonctions ci-dessous:

```
// fonction qui vérifie si le caractère est une lettre int isLetter(char c);
int isDigit(char c);
// fonction qui vérifie si un nombre est un nombre pair
int isEven(int value);
// fonction qui vérifie si un nombre est un nombre premier
int isPrime(int value);
// fonction qui calcule la somme des chiffres d'un nombre
int digitSum(int value);
// fonction qui calcule l'inverse d'un nombre
int reverseNumber(int value);
// fonction qui convertit les lettres minuscules en majuscules
void upperCase(char c);
// fonction qui convertit les majuscules en minuscules
void downcase(char c);
// fonction pour vérifier si le caractère est en majuscule
int isUpperCase(char c);
// fonction factorielle
int factoriel(int value);
// fonction puissance
int power(int base, int degree);
// fonction qui calcule les racines d'une équation quadratique à une inconnue
void findRoots(int a, int b, int c);
// fonction qui calcule si un nombre est un palindrome
int isPalindromeNumber(int value);
// fonction qui calcule si le texte est palindrome
int isPalindromeText(char *c, int size);
// fonction qui vérifie si le caractère est une voyelle
int isVowel(char c);
```

```

// fonction qui calcule si ce caractère est dans le tableau
int exist(int *arr, int size, int value);
// fonction qui calcule la première place du caractère recherché, s'il y en a un dans le tableau
int getIndex(int *arr, int size, int value);
// fonction qui calcule combien de fois le caractère recherché apparaît dans le tableau
int count(int *arr, int size, int value);
// fonction qui calcule le nombre maximum de tableaux
int findMax(int *arr, int size);
// fonction qui calcule le 2ème max du tableau
int findMax2(int *arr, int size);
// fonction qui calcule le nombre min du tableau
int findMin(int *arr, int size);
// fonction qui calcule le 2ème nombre min du tableau
int findMin2(int *arr, int size);
// fonction qui calcule la somme des éléments du tableau
int sumArray(int *arr, int size);
// fonction qui calcule la moyenne du tableau
double averageArr(int *arr, int size);

```

IMPORTANT:

- 1) Vous devez effectuer des études TP en tant que « fichier d'en-tête – fichier source – fichier de test » : écrivez les en-têtes des fonctions que vous avez écrites dans le fichier d'en-tête, les codes des fonctions dans le fichier source et les codes de test dans le fichier d'essai.
Pour la partie 1, créez un fichier bibliothèque distinct .h et .c pour chaque structure dans tp01.c pour toutes les méthodes demandées, tp01.h pour les titres, test.c pour les codes de test. Vous pouvez tester dans le fichier test.c que vous avez créé pour la partie 1.
- 2) Nous préférons faire des études de TP dans un environnement linux, sans avoir besoin d'IDE supplémentaire.
- 3) Compressez vos fichiers et téléchargez-le sur le système en les nommant « NumeroEtudiant_Prenom_Nom_TPX.zip ».