

# Algorithmique et Structures de Données

## TP 0 - Rappel Notions de Base en langage C

Halim Djerroud (hdd@ai.univ-paris8.fr)

### == Les structures de contrôle

#### Exercice 1 :

1. Écrire un programme qui affiche les 100 premiers nombres pairs, pour chaque nombre  $n$  de  $[1..100]$  vérifier si le  $n$  est pair. Améliorer votre programme pour n'afficher que les nombres pairs.
2. Écrire un programme qui affiche les 100 premiers nombres premiers ( $n$  est premier si il est divisible par 1 et par lui même).
3. Écrire un programme qui permet de lire une suite de chiffres au clavier (la saisie se termine quand l'utilisateur saisi un nombre négatif) et détermine le minimum et le maximum de ces chiffres, les chiffres saisis ne sont pas tous stockés en mémoire.
4. Écrire un programme *calc* qui permet de simuler une calculatrice (opérateurs supportés : +, -, \*, / ), le programme n'est pas interactif l'utilisateur doit lui indiquer les valeurs à calculer en argument du programme, par exemple :  
*./calc 2 + 5*

#### Exercice 2 :

1. Écrire un programme qui affiche le triangle de pascal jusqu'à  $n$  ( $n$  saisi au clavier).  
1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1  
...  
2. À quoi sert le triangle de pascal.
3. Écrire un programme qui lit une suite de valeurs au clavier (la saisie se termine quand l'utilisateur saisi un nombre inférieur à 0) et affiche la variance et l'écart-type.

$$V =$$
$$\sigma = \sqrt{V}$$

4. Que signifie la variance ? Que signifie l'écart-type ?
5. Écrire un programme qui affiche la barre de progression suivante (voir image), les segments (vertical, diagonal, horizontal) sont affichés à tour de rôle, avec un délai d'1 seconde, utilisez un tableau à deux dimensions pour représenter l'image.

```

      *      *
    *      *
  *      *
*      *
* * * * *
  *      *
    *      *
      *      *

```

6. Écrire un programme qui demande à l'utilisateur de saisir deux matrices carrées, calculer et afficher leurs somme et leurs produit. Remarque : la taille des matrices est déterminée par l'utilisateur en argument lors du lancement du programme.

## == Les pointeurs

### Exercice 3 :

1. Écrire un programme qui déclare 3 variables et affiche leurs adresses et leurs contenus.
2. Écrire une fonction qui prend en paramètres deux variables et échange leurs contenu.
3. Écrire une fonction qui a comme paramètres un tableau d'entiers de taille quelconque, la taille du tableau est donnée en paramètre, et 2 pointeurs vers des entiers min et max. La fonction doit renvoyer dans les entiers pointés par min et max respectivement les plus petits et les plus grands entiers du tableau.
4. Écrire un programme qui permet de lire une chaîne de caractère saisie par l'utilisateur et compte le nombre de caractères de cette chaîne.
5. Écrire un programme qui permet de lire une chaîne de caractère saisie par l'utilisateur et compte le nombre de majuscules et de minuscules.

## == Les structures

### Exercice 1 :

1. Écrire des fonctions de lecture, d'écriture et d'affichage d'une variable de type Date. La Date est stocké dans la structure suivante :

```

typedef struct {
    int jour;
    char mois[10];
    int annee;
} DATE;

```

2. Écrire un programme qui permet de stocker les informations des étudiants dans un tableau. Le nombre maximal d'étudiants est de 10. Les informations à stocker sont les suivantes : Id, Nom, Prénom, Groupe, Note.
  - Afficher un menu interactif qui permet à l'utilisateur de choisir parmi les fonctionnalités suivantes : 1) Afficher la liste des étudiants et leur moyenne, 2) Ajouter un étudiant, 3) Supprimer un étudiant, 4) Modifier un étudiant, 5) Quitter le programme.

#### Exercice 4 :

1. Représenter un nombre complexe, définir un type de structure qui contient deux flottants, la partie réelle et la partie imaginaire.
2. Écrire une fonction permettant de saisir un complexe passé en paramètre
3. Écrire une fonction permettant d'afficher un complexe passé en paramètre
4. Écrire une fonction d'addition qui reçoit en paramètre deux complexes, qui fait la somme des deux et range le résultat dans le premier

## == Les tableaux

#### Exercice 5 :

1. Écrire un programme qui permet à l'utilisateur de saisir 10 valeurs entières et les afficher.
2. Améliorer le programme (un ou plusieurs fonctions) afin qu'il puisse saisir un nombre de valeur choisi par l'utilisateur.
3. Trier les valeurs du tableau avec un des algorithmes suivants : Tri par sélection, tri par insertion, tri à bulles, tri rapide, tri par tas ou tri par fusion. Vous devez écrire une fonction qui renvoie un nouveau tableau.
4. Écrire une fonction qui permet de rechercher une valeur.
5. Remplacer les valeurs entières par la structure suivante

```
typedef struct {
    char* nom;
    char* prenom;
    float note;
} Etud;
```

6. adapter les fonctions précédentes (tri et recherche)

## == Les listes chaînées

#### Exercice 6 :

1. Écrire un programme qui utilise une liste doublement chaînée pour stocker une liste d'étudiants (structures Etud définie dans exercice précédent)

2. Écrire une fonction qui permet de trier la liste chaînée.
3. Écrire une fonction qui permet de recherche un élément.