# sciences informatiques

# cours et TP (60+60 H)

## Objectifs

Au terme de cette matière, l’étudiant devrait être capable de :

– Reconnaître les caractéristiques d’un micro-ordinateur.

– Savoir décomposer un problème pour mieux le résoudre.

– Pouvoir passer d’un problème à sa solution algorithmique.

– Etablir un algorithme juste, efficace et rapide.

– Savoir traduire un algorithme établi en un programme correct.

– prendre conscience progressivement des principes de recherche, d’utilisation et de complexité des algorithmes;

– acquérir les principes fondamentaux qui permettent d’analyser et de programmer des problèmes concrets;

– apprendre le processus de conception structurée, apprendre l’écriture d’un programme en langage algorithmique afin de le traduire en langage de programmation;

– adapter un algorithme à différentes structures de données.

## Contenu

Première partie : Algorithme (20 H)

Chapitre 1 : ARCHITECTURE et fonctionnement du matériel informatique

1.1 Introduction

1.1.1 Intérêts de l’informatique.

1.1.2 Quelques définitions :

1.1.2.1 l’information et ses représentations.

1.1.2.2 l’informatique.

1.1.2.3 l’ordinateur : schéma général d’un micro-ordinateur.

1.2 Les éléments d’un micro-ordinateur

1.2.1 L’unité centrale :

1.2.1 Processeur : description, rôle, exemples de processeurs.

1.2.2 Horloge : rôle.

1.2.3 Mémoire centrale : types, rôle.

1.2.4 Mémoire cache.

1.2.5 Bus : types, rôle.

1.2.6 Les mémoires auxiliaires :

1.2.6.1 Disque dur : description, principe de fonctionnement, avantages et inconvénients.

1.2.6.2 Disquette : description, principe de fonctionnement, lecteur, avantages et inconvénients.

1.2.6.3 Compact Disc (CD) : description, principe de fonctionnement, lecteur, avantages et inconvénients.

1.2.6.4 Dernières nouveautés.

1.2.7 Adaptateur d’entrée/sortie : types, rôle.

1.2.8 Les connexions (interfaçage : sériel, parallèle, USB, SCSI, …)

1.2.9 Les périphériques d’entrée :

1.2.9.1 Clavier : groupement et fonctions des touches.

1.2.9.2 Souris : rôle et fonctionnement.

1.2.9.3 Lecteur optique : rôle et fonctionnement.

1.2.9.4 Scanner : rôle et fonctionnement.

1.2.9.5 Dernières nouveautés.

1.2.10 Les périphériques de sortie :

1.2.10.1 Ecran : types et fonctionnement.

1.2.10.2 Imprimantes : types et fonctionnement.

1.2.10.3 Table traçante : rôle et fonctionnement.

1.2.10.4 Dernières nouveautés.

Chapitre 2 : Description et fonctionnement des logiciels

2.1 Logiciels de base

2.1.1 Introduction au système d’exploitation (Ex. Windows) : Gestion des ressources - éditeur de liens - chargeur.

2.1.2 Traducteur : assembleur, compilateur, interpréteur, macro générateur

Chapitre 3 : Introduction générale à l’algorithmique et à la programmation

3.1 Définition et concepts de base.

3.2 La notion de codage et d’instruction.

3.3 La notion de variable (objets mutables et affection de base).

3.4 De l’algorithme au programme (spécification, programme).

3.5 Les types de base.

3.6 Les opérateurs (logiques, arithmétiques, de relation, …).

3.7 Les expressions (logiques, arithmétiques, …).

3.8 Les paramètres.

3.9 Exemples et applications.

Chapitre 4 : Structure de l’algorithme et organigramme

4.1 Séquences.

4.2 Alternativité (si … alors … sinon)

4.3 Sélection et choix (cas)

4.4 Itérativité (tant que, pour, répéter, récursivité, …)

4.5 Exemples et exercices appliqués en un logiciel de choix.

Chapitre 5 : Structures imbriquées

5.1 Alternatives imbriquées (si … alors … si … alors … sinon … sinon,).

5.2 Boucles imbriquées.

5.3 Structures complexes :

5.3.1 Décomposition.

5.3.2 Introduction et utilisation de : fonction et procédure

5.3.3 Itérativité et récursivité.

5.4 Exemples et exercices appliqués en un logiciel de choix.

Chapitre 6 : Les tableaux

6.1 Tableaux à une dimension

6.1.1 Présentation en mémoire et vocabulaire.

6.1.2 Parcours et recherche.

6.1.3 Insertion et suppression.

6.1.4 Permutation et triage.

6.1.5 Exemples et exercices appliqués en un logiciel de choix.

6.2 Tableaux à deux dimensions

6.2.1 Présentation en mémoire.

6.2.2 Parcours et recherche.

6.2.3 Insertion et suppression.

6.2.4 Transfert, permutation et triage.

6.2.5 Exemples et exercices appliqués en un logiciel de choix.

## deuxieme partie : Langage C (40 H)

## Objectifs

L’enseignement de ce cours doit :

– Décrire la méthode de développement par raffinages successifs

– Permettre aux étudiants d’acquérir les concepts de l’analyse algorithmique.

– Permettre aux étudiants d’acquérir les concepts de la programmation en C en vue d’une utilisation dans le domaine de l’informatique industrielle.

## Contenu

Chapitre 1   
Introduction

1.1.1 La programmation modulaire et structurée : notion de projet, atelier de développement et groupes de travail, modules et compilation séparée

1.1.2 Historique du langage C et ses points forts

1.1.3 Structure d’un programme C : présentation rapide, exemples

Chapitre 2  
Les aspects classiques

1.2.1 Les éléments de base : l’alphabet du langage, les items syntaxiques (identificateurs, mots-clés, littéraux, opérateurs, séparateurs), l’alphabet du langage

1.2.2 Les déclarations : les types scalaires arithmétiques, initialisation des variables scalaires arithmétiques, les tableaux unidimensionnels et bidimensionnels

1.2.3 Les opérateurs (arithmétiques, relationnels, logiques, d’affectation, les opérateurs combinés à l’affectation, de choix, virgule, d’adressage, les opérateurs sur les chaînes de bits, sizeof)

1.2.4 Les expressions

1.2.5 Objets, gvaleurs, valeurs, la priorité des opérateurs, les conversions (implicite et explicite)

1.2.6 Les instructions (nulle, d’affectation, l’instruction expression, de choix, de boucles (while, do, for), le bloc, l’instruction switch, l’instruction break, l’instruction continue)

1.2.7 Les entrées/sorties élémentaires

1.2.8 Les entrées/sorties de caractères : la fonction putchar, la fonction getchar

1.2.9 Les entrées/sorties formatées : la fonction printf, la fonction scanf

Chapitre 3  
Les sous-programmes

1.3.1 Définition d’une fonction (entête de la fonction, corps de la fonction), exemples

1.3.2 Appel de fonction, passage de paramètres

1.3.3 Les variables locales : la validité des variables locales, les attributs d’implémentation

1.3.4 Les variables globales

1.3.5 La récursivité

Chapitre 4  
Structures et unions

1.4.1 Déclaration de structures, initialisation de structures, accès à un champ

1.4.2 Utilisation des structures : composition des structures, tableaux de structures, pointeurs de structures, structures récursives

1.4.3 Les unions

Chapitre 5  
Pointeurs et tableaux

1.5.1 Pointeurs et adresses

1.5.2 Opérations sur les pointeurs : la valeur NULL, affectation de pointeurs, incrémentation de pointeurs, comparaison de pointeurs, soustraction de pointeurs, affectation de chaînes de caractères

1.5.3 Gestion dynamique de la mémoire

1.5.4 Pointeurs et tableaux

1.5.5 Passage de paramètres

1.5.6 Tableaux de pointeurs

1.5.7 Pointeurs de fonctions

Chapitre 6  
Compléments

1.6.1 Le préprocesseur

1.6.2 Les macros définitions : les macros définitions paramétrées, destruction de macros définition (undef)

1.6.3 Inclusion de fichier source: #if, #ifdef, #ifndef

1.6.4 La compilation conditionnelle

1.6.5 Principes de la programmation modulaire : communication entre modules, règles de communication, protection de l’information

1.6.6 Le surnomme de type (typedef)

1.6.7 Les expressions statiques

1.6.8 Déclaration d’énumérations : représentation des valeurs, utilisation des énumérations

Chapitre 7  
La bibliothèque standard

1.7.1 Les entrées/sorties : généralités, les entrées/sorties standards, les entrées/sorties formatées, les entrées/sorties de lignes de caractères

1.7.2 La manipulation des fichiers : ouverture d’un fichier, la fermeture d’un fichier, les entrées/sorties de caractères, les entrées/sorties formatées, les entrées/sorties de lignes de caractères, les entrées/sorties de blocs d’informations