

INTRODUCTION GENERALE

(Rappel)

Objectifs:

① Rappeler l'étudiant les notions de base concernant l'informatique et l'algorithmiques déjà connues au niveau précédent.

② Mettre la programmation dans son contexte.

Pré requis:

① Informatiques générale

② Architecture des ordinateurs

③ Algorithmique

Plan

I/ INTRODUCTION	2
1. Définition de l'informatique	
2. la composante matérielle	
3. la composante logicielle	
II/ LES ETAPES DE RESOLUTION D'UN PROBLEME	3
1. la définition exacte du problème	
2. La programmation	
3. La formalisme du problème	
III/ LES CONCEPTS DE LA PROGRAMMATION	4
1. Les programmes	
2. Les structures	
3. INTRODUCTION AU LANGAGE C	5

INTRODUCTION GENERALE

(Rappel)

I/ INTRODUCTION

1. Définition de l'informatique

L'informatique est le traitement automatique de l'information. Pour automatiser le traitement de l'information, on a besoin d'un système informatique, c'est l'ordinateur. Un ordinateur est un ensemble de deux composantes, la composante matérielle et la composante logicielle.

la composante matérielle

Elle constitue la partie matérielle " physique " de l'ordinateur. on l'appelle souvent "HARD WERE".

Un ordinateur individuel comporte, dans sa configuration la plus simple, une unité centrale, une mémoire principale, des terminaux (écran, clavier, souris ...), une mémoire secondaire et une imprimante.

2. la composante logicielle

Elle constitue la partie logique de l'ordinateur. on l'appelle souvent "SOFT WERE". Cette composante est divisée en deux catégories (logiciels de base et logiciels d'application). Dans les logiciels de base on trouve le système d'exploitation et les utilitaires. Dans ces derniers on trouve les langages de programmation et d'autres utilitaires nécessaires à la conception des logiciels d'application.

Les logiciels d'application représentent le fruit du système informatique. Ils répondent aux besoins de l'utilisateur.

II/ LES ETAPES DE RESOLUTION D'UN PROBLEME

La tâche la plus importante dans l'informatisation d'un problème réel est la phase de transformation des objets réels en objets logiques. Cette transformation nécessite trois étapes:

- ✓ la définition exacte du problème
- ✓ la formalisation du problème
- ✓ la programmation

1. la définition exacte du problème

Pour résoudre et automatiser un problème avec un ordinateur, il est nécessaire de le définir et de préciser les données qu'on dispose et les objectifs qu'on souhaite atteindre.

Exemple: Si on désire résoudre une équation de second degré ($AX^2 + BX + C = 0$) à l'aide d'un ordinateur, la définition du problème doit comporter les réponses aux questions suivantes:

* Quels sont les données à fournir à l'ordinateur?

De quel type sont-elles?

Et sous quelles formes doit-on les introduire?

* Que va faire l'ordinateur si les données fournies ne forment pas une équation de second degré?

* Que faire si DELTA est négatif?

* Sous quelles formes va-t-on afficher le ou les solutions. ... etc.

D'une façon générale, la définition d'un problème consiste à prévoir des réponses à tous les cas envisageables .

2. La formalisme du problème

C'est la phase algorithmique. Elle fournit la méthode et la démarche que l'ordinateur va suivre pour résoudre le problème posé. La phase la plus difficile de la résolution d'un problème n'est pas la traduction de l'algorithme dans n'importe quel langage, mais c'est la conception de l'algorithme lui-même. Cette étape est connue sous le nom d'analyse.

3. La programmation

C'est le passage du modèle logique qui est un algorithme, au modèle directement exploitable par l'ordinateur. Pour cela, il faut traduire l'algorithme en un programme écrit dans un langage de programmation choisis par l'utilisateur. Le langage de programmation dépendra de la nature du problème ou de l'application à automatiser.

La programmation est donc, l'art de concevoir et d'exprimer des algorithmes.

III/ LES CONCEPTS DE LA PROGRAMMATION

1. Les programmes

Tout programme est une session d'instructions (ordres) qui a les caractéristiques suivantes:

- * Exécution séquentielle, sauf indication explicite.
- * Un traitement à effet. Souvent un impression, un affichage de caractères à l'écran ou une action sur des objets (données, variables,...etc).
- * Les objets (variables et données) sont toujours déclarés avant les instructions.
- * Il est possible qu'il soit nécessaire d'exécuter une instruction ou un groupe d'instructions plus qu'une fois.
- * Le programme lui même est une entité statique, le traitement est dynamique.

Un programme est un moyen par lequel le programmeur communique avec le processeur.

2. Les structures

Tout problème d'algorithmique peut être résolu en utilisant quatre structures de bases qui sont:

- ✓ la séquence
- ✓ la décision
- ✓ la répétition
- ✓ les procédures (modularité et abstraction).

* une séquence est un groupe d'instructions exécutés l'une après l'autre.

* une décision est une structure permettant à une action du programme d'être influencée par les données.

Dans beaucoup de langages la structure décision commence par le mot **if**, ce qui donne des instructions de genre:

```
If(X>=0)
    Y=X
else Y=-X;
ce qui est équivalent à :
Si X>=0 alors
    Y←X
Sinon Y←-X
Fin si
```

* La structure de boucle est utilisée pour exécuter une instruction ou une suite d'instructions plusieurs fois à la fois alors que les instructions sont les mêmes chaque fois que la boucle est exécutée. Les données sur lesquelles elles agissent changent d'une

fois à l'autre. Par exemple: Si on répète cent fois l'instruction $X \leftarrow X+1$, l'effet est d'additionner 100 à X.

* La procédure nous permet de remplacer un groupe d'instructions par un nom qui identifie ce groupe.

Les procédures permettent de donner au programmes une structure modulaire, ce qui rend d'écriture plus facile. C'est un mécanisme qui facilite l'abstraction.

IV/ INTRODUCTION AU LANGAGE C

Soit le programme suivant:

```
#include <stdio.h>
#include <math.h>
main()
{
    printf("%f",sqrt(2));
}
```

Si on exécute ce programme, l'ordinateur affiche sur l'écran $1.4142356 \cong \sqrt{2}$

"**printf**" est le cœur de ce programme dont le résultat est l'affichage de la valeur de $\text{sqrt}(2)$, qui est son argument. "**sqrt**" est l'une des nombreuses fonctions que fournit la bibliothèque des fonctions mathématique "**math.h**" du langage C.

Soit le deuxième programme:

```
#include <stdio.h>
#include <math.h>
main()
{
    int X;
    scanf("%d",&X);
    if(X>=0)
        printf("la racine carrée de %d est %f",X,sqrt(X));
    else
        printf("pour %d pas de racine carrée réelle ",X);
}
```