DEUG MASS Info1 – TD 4 (Lundi 31 octobre à vendredi 4 novembre 2005)

Objectifs de la séance:

1. boucles et tests.
2. compréhension/reconnaissance d'un algorithme.
3. programmation d'un jeu très simple.

Exercice 1: soit le programme « mystere.c » suivant:

```
#include <stdio.h>
main()
{ int n,e;
  float x,y,p;
  scanf("%f %d",&x,&n);
  y=x; e=n; p=1.;
  while (e>0)
  { if (e%2 ==1) p*=y;
    e/=2; y*=y;
  }
  printf("nombre_mystère(%f, %d) = %f\n",x,n,p);
}
```

question 1.1q: Que font les instructions suivantes:

e/=2;
y*=y;
p*=y;

Récrivez-les différemment.

question 1.2: Quelle est la nature de l'expression (e%2 = = 1) ?
Que vaut-elle en fonction de la valeur de e ?
Que ferait l'instruction: if (e%2) p*=y;

question 1.3: Que fait le programme ?
Aide: au besoin faites-le tourner à la main sur un exemple simple (x=2, n=15 par exemple).

question 1.4: expliquer le fonctionnement de ce programme. On pourra, en particulier, donner une définition par récurrence du « nombre mystère », en rapport avec la conception de ce programme.

Exercice 2: Ecrire un programme qui lit une note sur 20, puis affiche la mention correspondante: "ajourné" si n<10, "passable" si n ≥ 10 et n < 12, "assez bien" si n ≥ 12 et n < 14, "bien" si n ≥ 14 et n < 16, "très bien" si n ≥ 16.

Exercice 3: Ecrire un programme qui joue au jeu suivant contre l'utilisateur du programme:
Au départ, il y a un tas de 50 allumettes (ou jetons, ou cailloux...). Chacun son tour, les 2 joueurs ôtent obligatoirement entre 1 et 6 allumettes. Celui qui ôte la dernière allumette gagne.
Bien sûr, il existe une stratégie gagnante, très simple, qu'il s'agit de trouver et de programmer, afin que le programme gagne automatiquement s'il hérite de son adversaire une position favorable.
Le programme affichera le nombre d'allumettes restant après les coups de chaque joueur (lui et l'utilisateur du programme).