UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Licence 1 INFO
INFO0204

TP n° 3 : Algorithmes classiques

Pour l'ensemble de ce sujet, vous aurez à développer des solutions de plus en plus sophistiquées : vous devez impérativement structurer vos codes, et y insérer tous les commentaires nécessaires pour vous y retrouver [et nous permettre de nous y retrouver].

1 Classique

1. `[ puissance.asm ]`
   On demande à l'utilisateur de saisir x et n et on affiche à l'écran la valeur de x^n.

2. `[ factorielle.asm ]`
   On demande à l'utilisateur de saisir n et on affiche à l'écran $n!$ (factorielle de n).

2 Moins classique...

`[ racine.asm ]`
Ce programme doit évaluer la racine carrée [entière] d'une valeur n saisie par l'utilisateur : il s'agit du plus grand entier [positif] tel que $a^2 \leqslant n$ (ce nombre existe car $n \geqslant 0$).
Exemple : si on saisit 9, 10, 11, 12, 13, 14 ou 15, la valeur affichée est 3.

3 Plus difficile

On souhaite écrire un programme qui affiche à l'écran la représentation binaire d'un entier saisi au clavier. L'affichage étant limité avec ce simulateur, les valeurs des bits seront affichées à raison d'un chiffre par ligne.

1. *affichage à l'envers* `[ binaire1.asm ]`
   Le programme demande à l'utilisateur de saisir un entier et affiche, dans l'ordre, les restes par des divisions par 2 successives.
   Exemple : si on saisit la valeur 5 les valeurs affichées doivent être, dans l'ordre : 1, 0, 1, 0, 0, 0, 0 et 0.

2. `[ binaire2.asm ]`
   Modifiez votre programme pour que l'affichage soit réalisé dans le bon ordre.
   Exemple : si on saisit 76 les valeurs affichées seront, dans l'ordre : 0, 1, 0, 0, 1, 1, 0 et 0.
   Indication : vous pouvez commencer par tester 128, puis 64, ...

3. Pour aller plus loin on peut se poser la question d'obtenir l'écriture binaire du nombre saisi (les 0 non significatifs ne doivent pas être affichés)
   Exemple : si on saisit 19 les valeurs affichées seront, dans l'ordre : 1, 0, 0, 1 et 1.

4 Vous en voulez encore ?

Malgré un affichage simplifié, on peut adapter en assembleur un certain nombre de programmes intéressants : si vous le souhaitez, faites vous plaisir ;-)

1. Nombre à deviner
   Le but du jeu est de deviner un nombre choisi dans un intervalle donné (ici [0 ;255]), en donnant des propositions successives. A chaque proposition, le programme indique au joueur si sa proposition est trop petite ou trop grande par rapport au nombre à deviner.
   (a) Ecrivez le programme assembleur : le nombre à deviner est placé dans la case mémoire n° 0 ; lorsque le joueur propose un nombre
   - si la proposition est trop petite, on affiche 0 à l'écran.
   - si la proposition est trop grande, on affiche 1 à l'écran.
   - si la proposition est juste, on affiche le résultat et le programme s'arrête.

   (b) Modifiez votre programme pour afficher le nombre de coups.
   (c) Modifiez votre programme pour limiter le nombre de coups.

2. Jeu de Nim
   Ce jeu se joue à deux joueurs. On part avec un certain nombre d'allumettes et les joueurs tirent, à tour de rôle, de 1 à 3 allumettes. Le perdant est celui qui tire la dernière allumette.
   (a) Ecrivez le code assembleur :
   - le nombre initial d'allumettes est placé dans la case n° 0 de la mémoire de données
   - le programme affiche le numéro du joueur qui va devoir jouer (1 ou 2)
   - on affiche ensuite le nombre d'allumettes restantes...
   - puis on demande un nombre d'allumettes à enlever
   - le joueur saisit alors un entier entre 1 et 3 [mais au plus égal au nombre d'allumettes restantes]
   - si le choix est invalide (0 ou strictement supérieur à 3, ou encore trop grand par rapport au nombre d'allumettes restantes), on demande au joueur de faire un nouveau choix.

   (b) Proposez une solution pour jouer contre l'ordinateur. [1]

1. https://www.youtube.com/watch?v=k0HD8ASx_xw
   https://www.youtube.com/watch?v=168oiOnTqFE