

TD2

SI4 – Base de la programmation

Ce travail porte sur la mise en œuvre des compétences acquises en programmation sur machine. Cette fois il faut coder un algorithme sur papier.

Dans ce cas, les exercices réalisés peuvent être écrits en pseudo-code. Vous avez à votre disposition les outils vus en cours (arbres programmatiques, algorithmes). Pour corriger le programme nous utiliserons un traçage à l'aide d'un jeu d'essai.

- 1.1 Ecrire une petite application qui permet de saisir un chiffre entier pour pouvoir ensuite le deviner avec des indices « plus grand » et « plus petit ». L'application doit pouvoir relancer une nouvelle partie. Il faut également compter le nombre de coups nécessaires pour gagner.
- 1.2 Ecrire un jeu d'essai afin de valider le comportement du programme. Ce jeu d'essai associe le contenu des variables du programme à chaque instruction traitée.
- 1.3 Modifier l'application précédente en introduisant la notion de fonction. Une fonction « chiffreAléatoire » permet de délivrer un nombre pseudo-aléatoire.

```
FONCTION chiffreAléatoire(borneInférieure : entier, borneSupérieure entier) :entier
// borneInférieure et borneSupérieure sont incluses.
```

- 2.1 Ecrire une petite application qui permet de réaliser le fameux jeu du pendu. L'utilisateur saisit un mot, ensuite, un joueur devine la première lettre de ce mot. Une fois la lettre trouvée, il cherche la suivante et ainsi de suite. Chaque coup est comptabilisé, afin de produire un score. La partie peut être relancée. On peut aussi planifier un nombre de coups maximum, qui arrête le jeu s'il est dépassé.
- 2.2 Ecrire un jeu d'essai afin de valider le comportement du programme. Comme pour la question 1.2, faire un tableau.
- 2.3 Modifier le jeu du pendu en ajoutant la fonction « motAléatoire ».

```
FONCTION motAléatoire(Langue : entier) : chaîne de caractères[255]
// Anglais : 1, Français : 2, Espagnol : 3, Allemand : 4
```

Si l'écriture d'un programme vous bloque, passer par la solution littérale. C'est à dire qu'il faut décomposer les grandes étapes du programme avec des phrases. Chaque phrase peut ensuite être décomposée en phrases plus petites jusqu'à obtenir des opérations (phrases, mots) indivisibles. Essayez également de projeter les opérations d'entrées et sorties.

