Université Paul Cézanne
Licence Sciences et Technologies – Mention Mathématique et Informatique (M.I.)
I1 : Algorithmique et Programmation

TD 3 : Instructions répétitives

- ***Boucles simples***

1. Ecrire un programme C qui calcule la somme et le produit des N premiers entiers, en utilisant l'instruction while.

2. Ecrire un programme C qui réalise la saisie d'un entier N et détermine s'il est premier ou non. Ceci en utilisant l'instruction while

3. Soit le programme C suivant :

```
main()
{
    int  i;
    for (i = 0 ; i < 10 ; i = i + 1)
      printf("carre de %d = %d \n", i , i * i);
}
```

3.1. Réécrire un programme équivalent à celui-ci mais avec l'instruction **while**.
3.2. Même question avec l'instruction **do….while**

4. Peut-on réécrire les programmes des exercices 1. et 2. avec l'instruction do….while ou l'instruction for à la place du while ? Si oui, réécrire le programme. Expliquer laquelle des 3 instructions est la plus appropriée dans chaque exercice.

- ***Suites récurrentes***

5. Ecrire un programme C qui, étant donné un entier n >0 , calcule la suite F définie comme suit:

$$F = 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{n}$$

6. Ecrire un programme C qui, étant donné un entier $n \geq 0$, calcule le terme d'ordre n (noté U_n) de la suite définie par la relation de récurrence suivante :

$U_0 = 2$ et $\forall n \geq 1, U_n = 3 * U_{n-1} - 2$

7. Ecrire un programme C permettant de calculer le terme d'ordre n de la suite de Fibonacci définie par la relation de récurrence suivante : :

$U_0 = 0, U_1 = 1$, et $\forall n \geq 2 , U_n = U_{n-1} + U_{n-2}$

8. Ecrire l'algorithme itératif permettant de calculer une valeur approchée (à un epsilon prés) de racine n-ième d'une valeur a, par la suite récurrente suivante:

$$\begin{cases} U_0 = \dfrac{a}{n} \\ U_{k+1} = \dfrac{\left(U_k^n (n-1) + a\right)}{n.U_k^{n-1}} \end{cases}$$

Critère d'arrêt :

$$\left|a - U^n\right| < \varepsilon$$

9. Ecrire un programme C qui, étant donné 2 entiers n ≥ p ≥ 0, calcule : $C_n^p = \dfrac{n!}{(n-p)!\,p!}$