



Programmation en C et C++

Faculté des Sciences, de Génie et d'Architecture (FSGA)

Devoir #1

A réaliser individuellement

La date limite de remise est fixée au **Samedi 14 Mars 2019**

Note1 : Les fichiers (document en PDF, fichiers sources et exécutables) du devoir doivent être postés au portail Web du cours jusqu'au **Samedi 14 Mars 2019 à 23h59mn59sec.** Passé ce délai, aucun devoir ne sera admis.

Note2 : Aucun devoir ne sera reçu par mail, la note **0.00** sera accordée automatiquement à tout étudiant n'ayant pas remis leur devoir.

Note3 : Le fichier Ms. Word et le code du programme doivent être compressé dans un fichier **.zip** ou **.rar** et posté au portail du cours. Pour info sur comment compresser des fichiers sur Windows, regardez cette vidéo :

Réf. <https://www.youtube.com/watch?v=hvllEkxJPrU>)

| ■ Identification Étudiant | |
|---------------------------|--|
| Nom : | |
| Prénom(s) : | |
| Faculté : | |
| Option : | |

Exercice 1 : Programmation modulaire 1 (en C)

La capitalisation consiste à faire avancer dans le temps une valeur présente pour calculer sa valeur future appelée aussi Valeur Acquise. Ainsi, la valeur future ou acquise S_n d'une somme d'argent présente S_0 disponible après n années et placée au taux t est égale à : $S_n = S_0(1 + t)^n$

Une somme de S_0 gourdes est placée pendant n années au taux annuel de $t\%$

1. La valeur acquise à l'issue de ce placement

Formule à utiliser : $S_n = S_0(1 + t)^n$

2. Si au bout de cette période de placement on souhaite obtenir S_n gourdes, Quelle somme doit-on placer aujourd'hui ?

Formule à utiliser : $S_n = S_0(1 + t)^n \Rightarrow S_0 = S_n(1 + t)^{-n}$

3. Si la somme placée aujourd'hui est de S_0 gourdes, après combien de temps disposera-t-on d'une somme égale à S_n gourdes ?

Formule à utiliser

$$S_n = S_0(1 + t)^n \Rightarrow \log[S_n] = \log[S_0(1 + t)^n] \Rightarrow n = \frac{\log S_n + \log S_0}{\log(1 + t)}$$

4. Si au bout de n années la valeur acquise du placement est de S_n gourdes à quel Taux le placement a été effectué ?

Formule à utiliser

$$S_n = S_0(1 + t)^n \Rightarrow (1 + t)^n = \frac{S_n}{S_0} \Rightarrow t = \left(\frac{S_n}{S_0}\right)^{\frac{1}{n}} - 1$$

A faire :

Ecrire en C un programme **capitalisation.c** qui :

1. Présente un menu à l'utilisateur lui demandant de faire le choix de l'opération à exécuter parmi les suivantes :
 - Valeur acquise
 - Placement initial
 - Terme du prêt
 - Intérêt de placement
2. Permettre d'effectuer l'opération en saisissant les paramètres nécessaires en fonction du choix de l'utilisateur.

NB1 : Pour chaque type d'opération, vous devez créer une fonction.

NB2 : Fonctions additionnelles à utiliser :

Bibliothèque : **<math.h>**

Fonctions :

- **log(x)**, pour le logarithme
- **pow(x,n)** pour la puissance

| |
|---------------------|
| ■ Démarche utilisée |
|---------------------|

Ici, vous devez décrire en français ou en anglais la démarche utilisée pour réaliser le programme

■ Écrans d'exécution de l'exercice

Ici, vous devez présenter deux (2) écrans d'exécution du programme.

Exercice 2 : Programmation modulaire 1 (en C++)

La capitalisation consiste à faire avancer dans le temps une valeur présente pour calculer sa valeur future appelée aussi Valeur Acquise. Ainsi, la valeur future ou acquise S_n d'une somme d'argent présente S_0 disponible après n années et placée au taux t est égale à : $S_n = S_0(1 + t)^n$

Une somme de S_0 gourdes est placée pendant n années au taux annuel de $t\%$

5. La valeur acquise à l'issue de ce placement

Formule à utiliser : $S_n = S_0(1 + t)^n$

6. Si au bout de cette période de placement on souhaite obtenir S_n gourdes, Quelle somme doit-on placer aujourd'hui ?

Formule à utiliser : $S_n = S_0(1 + t)^n \Rightarrow S_0 = S_n(1 + t)^{-n}$

7. Si la somme placée aujourd'hui est de S_0 gourdes, après combien de temps disposera-t-on d'une somme égale à S_n gourdes ?

Formule à utiliser

$$S_n = S_0(1 + t)^n \Rightarrow \log[S_n] = \log[S_0(1 + t)^n] \Rightarrow n = \frac{\log S_n + \log S_0}{\log(1 + t)}$$

8. Si au bout de n années la valeur acquise du placement est de S_n gourdes à quel Taux le placement a été effectué ?

Formule à utiliser

$$S_n = S_0(1 + t)^n \Rightarrow (1 + t)^n = \frac{S_n}{S_0} \Rightarrow t = \left(\frac{S_n}{S_0}\right)^{\frac{1}{n}} - 1$$

A faire :

Ecrire en C++ un programme **capitalisation.cpp** qui :

- Présente un menu à l'utilisateur lui demandant de faire le choix de l'opération à exécuter parmi les suivantes :
 - Valeur acquise
 - Placement initial
 - Terme du prêt
 - Intérêt de placement
- Permettre d'effectuer l'opération en saisissant les paramètres nécessaires en fonction du choix de l'utilisateur.

NB1 : Pour chaque type d'opération, vous devez créer une fonction.

NB2 : Fonctions additionnelles à utiliser :

Bibliothèque : **<math.h>**

Fonctions :

- **log(x)**, pour le logarithme
- **pow(x,n)** pour la puissance

| |
|---------------------|
| ■ Démarche utilisée |
|---------------------|

Ici, vous devez décrire en français ou en anglais la démarche utilisée pour réaliser le programme

■ Écrans d'exécution de l'exercice

Ici, vous devez présenter deux (2) écrans d'exécution du programme.

Exercice 3 : Programmation modulaire 2 (en C)

1. Écrire une fonction **isPremier(n)** qui teste si un nombre **n** (compris entre **1** et **50**) et saisi par l'utilisateur est premier ou pas. Si le nombre saisi est en dehors de cet intervalle, la fonction doit forcer l'utilisateur à faire un autre choix, c'est-à-dire un nombre compris entre **1** et **50** inclusivement.
2. Écrire une fonction **nbPremier(val₁, val₂)** qui détermine et affiche tous les nombres premiers compris entre **val₁** et **val₂**.
val₁ et **val₂** seront saisies par l'utilisateur avec pour contraintes :
 - $001 \leq val_1 \leq 200$
 - $500 \leq val_2 \leq 1000$

Écrire un programme **nombrePremier.c** qui permettra à un utilisateur de manipuler les deux fonctions **isPremier(n)** et **nbPremier (val₁, val₂)** vérifier si un nombre est premier ou pour déterminer la liste des nombres premiers entre deux valeurs.

Le programme doit :

- Demander à l'utilisateur de saisir deux valeurs
- Déterminer tous les nombres premiers entre ces deux valeurs
- Afficher tous les nombres premiers entre ces deux valeurs
- Déterminer le nombre premier médian

NB : Un nombre premier est un nombre qui n'est divisible uniquement par 1 et par lui-même (1 est considéré comme premier).

| |
|---------------------|
| ■ Démarche utilisée |
|---------------------|

Ici, vous devez décrire en français ou en Anglais la démarche utilisée pour réaliser le programme

■ Écrans d'exécution de l'exercice

Ici, vous devez présenter deux (2) écrans d'exécution du programme.

Exercice 4 : Programmation modulaire 2 (en C++)

1. Écrire une fonction **isPremier(n)** qui teste si un nombre **n** (compris entre **1** et **50**) et saisi par l'utilisateur est premier ou pas. Si le nombre saisi est en dehors de cet intervalle, la fonction doit forcer l'utilisateur à faire un autre choix, c'est-à-dire un nombre compris entre **1** et **50** inclusivement.
2. Écrire une fonction **nbPremier(val₁, val₂)** qui détermine et affiche tous les nombres premiers compris entre **val₁** et **val₂**.
val₁ et **val₂** seront saisies par l'utilisateur avec pour contraintes :
 - 001 <= val₁ <= 200
 - 500 <= val₂ <= 1000

Écrire un programme **nombrePremier.cpp** qui permettra à un utilisateur de manipuler les deux fonctions **isPremier(n)** et **nbPremier(val₁, val₂)** vérifier si un nombre est premier ou pour déterminer la liste des nombres premiers entre deux valeurs.

Le programme doit :

- Demander à l'utilisateur de saisir deux valeurs
- Déterminer tous les nombres premiers entre ces deux valeurs
- Afficher tous les nombres premiers entre ces deux valeurs
- Déterminer le nombre premier médian

NB : Un nombre premier est un nombre qui n'est divisible uniquement par 1 et par lui-même (1 est considéré comme premier).

| |
|---------------------|
| ■ Démarche utilisée |
|---------------------|

Ici, vous devez décrire en français la démarche utilisée pour réaliser le programme

■ Écrans d'exécution de l'exercice

Ici, vous devez présenter deux (2) écrans d'exécution du programme.