

# Licence STS Université Claude Bernard Lyon1

## LIFAP1 – Séquence 1 – Groupes A, C, E, et G

### Sujet A

Contrôle Continu – TP6 - durée 40 min

Lundi 19 novembre 2018

#### Consignes (À lire impérativement avant de commencer !!!)

Vous devrez compiler et tester votre programme à chacune des étapes de son élaboration.

Vous donnerez à votre fichier source le nom suivant : TP6.cpp

Dans votre fichier, **vous mettrez en commentaire vos nom et prénom ainsi que votre numéro d'étudiant et le sujet (A ou B).**

Vous devrez obligatoirement utiliser les notions vues en TD / TP durant le semestre et rien d'autre !!!

La note tiendra compte :

- du respect des consignes ;
- de la qualité de la présentation et de la lisibilité du code ;
- des algorithmes ;
- du bon fonctionnement du programme.

Une fois le programme terminé et testé (ou à la fin du temps imparti), vous devrez déposer le fichier source (.cpp) via **TOMUSS** (en cliquant sur "déposer" dans la case Depot\_TP6 de l'UE LIFAP1).

#### Exercice

Soit la suite de Syracuse, définie comme suit :

$$u_{n+1} = 1 + 3u_n \text{ si } u_n \text{ impair ;}$$

$$u_{n+1} = u_n / 2 \text{ si } u_n \text{ pair.}$$

Quel que soit le nombre entier  $u_0$  strictement positif, cette suite finit par engendrer le nombre 1.

- Définir une constante MAX\_TAB ayant pour valeur 100.
- Écrire en C/C++ une **fonction** saisie\_valeur qui retournera au programme principal la valeur de  $n$  choisie par l'utilisateur. Attention, on recommencera la saisie **tant que la valeur n'est pas strictement positive**.
- Écrire en C/C++ une **fonction** syracuse qui remplit un tableau avec les termes de la suite de syracuse précédemment définie à partir d'une valeur lambda, donnée par l'utilisateur, et jusqu'à l'obtention de la valeur 1. Cette fonction devra retourner le nombre d'itérations au bout duquel la suite engendre la valeur 1. Exemple : Pour lambda= 11. Il faudra 14 itérations pour engendrer 1 et le tableau final sera :

11	34	17	52	26	13	40	20	10	5	16	8	4	2	1				
----	----	----	----	----	----	----	----	----	---	----	---	---	---	---	--	--	--	--

- Écrire en C/C++ la procédure affiche\_syracuse permettant l'affichage du tableau obtenu ; attention, on n'affichera que les valeurs significatives du tableau (i.e. celles pour lesquelles le tableau a été rempli).
- Écrire en C/C++ le programme principal permettant de saisir une valeur lambda, d'appeler la fonction syracuse et d'afficher le nombre d'itérations nécessaires pour converger vers la valeur 1, ainsi que le tableau obtenu.

# Licence STS Université Claude Bernard Lyon1

## LIFAP1 – Séquence 1 – Groupes A, C, E, et G

### Sujet B

Contrôle Continu – TP6 - durée 40 min

Lundi 19 novembre 2018

#### Consignes (À lire impérativement avant de commencer !!!)

Vous devrez compiler et tester votre programme à chacune des étapes de son élaboration.

Vous donnerez à votre fichier source le nom suivant : TP6.cpp

Dans votre fichier, **vous mettrez en commentaire vos nom et prénom ainsi que votre numéro d'étudiant et le sujet (A ou B).**

Vous devrez obligatoirement utiliser les notions vues en TD / TP durant le semestre et rien d'autre !!!

La note tiendra compte :

- du respect des consignes ;
- de la qualité de la présentation et de la lisibilité du code ;
- des algorithmes ;
- du bon fonctionnement du programme.

Une fois le programme terminé et testé (ou à la fin du temps imparti), vous devrez déposer le fichier source (.cpp) via **TOMUSS** (en cliquant sur "déposer" dans la case Depot\_TP6 de l'UE LIFAP1).

#### Exercice

Soit la suite de Fibonacci, définie comme suit :

$$u_0 = u_1 = 1 ;$$

$$u_{n+2} = u_{n+1} + u_n.$$

- Définir une constante `MAX_TAB` ayant pour valeur 20.
- Écrire en C/C++ une **fonction** `saisie_valeur` qui retournera au programme principal la valeur de `n` choisie par l'utilisateur. Attention, on recommencera la saisie **tant que la valeur n'est pas strictement positive et strictement inférieure à `MAX_TAB`.**
- Écrire en C/C++ une **procédure** `fibonacci` qui calcule et "renvoie" la valeur du  $n^{\text{ième}}$  terme de la suite.
- Écrire en C/C++ la procédure `remplit_tab` permettant de remplir le tableau avec les `n` premières valeurs de la suite, `n` étant passé en paramètre.
- Écrire en C/C++ la procédure `affiche_tab` permettant l'affichage du tableau obtenu ; attention, on n'affichera que les valeurs significatives du tableau (i.e. celles pour lesquelles le tableau a été rempli).
- Écrire en C/C++ le programme principal permettant de saisir une valeur `lambda`, et d'afficher les `lambda` premières valeurs de la suite en utilisant les sous-programmes précédents.

# Licence STS Université Claude Bernard Lyon1

## LIFAP1 – Séquence 1 – Groupes B, D, F et H

### Sujet A

Contrôle Continu – TP6 - durée 40 min

Lundi 19 novembre 2018

#### Consignes (À lire impérativement avant de commencer !!!)

Vous devrez compiler et tester votre programme à chacune des étapes de son élaboration.

Vous donnerez à votre fichier source le nom suivant : TP6.cpp

Dans votre fichier, **vous mettrez en commentaire vos nom et prénom ainsi que votre numéro d'étudiant et le sujet (A ou B).**

Vous devrez obligatoirement utiliser les notions vues en TD / TP durant le semestre et rien d'autre !!!

La note tiendra compte :

- du respect des consignes ;
- de la qualité de la présentation et de la lisibilité du code ;
- des algorithmes ;
- du bon fonctionnement du programme.

Une fois le programme terminé et testé (ou à la fin du temps imparti), vous devrez déposer le fichier source (.cpp) via **TOMUSS** (en cliquant sur "déposer" dans la case Depot\_TP6 de l'UE LIFAP1).

#### Exercice

La suite de Lucas est définie de la manière suivante :

$$\begin{aligned}\mathcal{L}_0 &= 2 \\ \mathcal{L}_1 &= 1 \\ \mathcal{L}_{n+2} &= \mathcal{L}_{n+1} + \mathcal{L}_n\end{aligned}$$

- Définir une constante `MAX_TAB` ayant pour valeur 30.
- Écrire en C/C++ une **fonction** `saisie_valeur` qui retournera au programme principal la valeur de `n` choisie par l'utilisateur. Attention, on recommencera la saisie **tant que la valeur n'est pas strictement positive et strictement inférieure à `MAX_TAB`.**
- Écrire en C/C++ une **procédure** `lucas` qui calcule et "renvoie" la valeur du  $n^{\text{ième}}$  terme de la suite.
- Écrire en C/C++ la procédure `remplit_tab` permettant de remplir le tableau avec les `n` premières valeurs de la suite, `n` étant passé en paramètre.
- Écrire en C/C++ la procédure `affiche_tab` permettant l'affichage du tableau obtenu ; attention, on n'affichera que les valeurs significatives du tableau (i.e. celles pour lesquelles le tableau a été rempli).
- Écrire en C/C++ le programme principal permettant de saisir une valeur `lambda`, et d'afficher les `lambda` premières valeurs de la suite en utilisant les sous-programmes précédents.

# Licence STS Université Claude Bernard Lyon1

## LIFAP1 – Séquence 1 – Groupes B, D, F et H

### Sujet B

Contrôle Continu – TP6 - durée 40 min

Lundi 19 novembre 2018

Consignes (À lire impérativement avant de commencer !!!)

Vous devrez compiler et tester votre programme à chacune des étapes de son élaboration.

Vous donnerez à votre fichier source le nom suivant : TP6.cpp

Dans votre fichier, **vous mettrez en commentaire vos nom et prénom ainsi que votre numéro d'étudiant et le sujet (A ou B).**

Vous devrez obligatoirement utiliser les notions vues en TD / TP durant le semestre et rien d'autre !!!

La note tiendra compte :

- du respect des consignes ;
- de la qualité de la présentation et de la lisibilité du code ;
- des algorithmes ;
- du bon fonctionnement du programme.

Une fois le programme terminé et testé (ou à la fin du temps imparti), vous devrez déposer le fichier source (.cpp) via **TOMUSS** (en cliquant sur "déposer" dans la case Depot\_TP6 de l'UE LIFAP1).

### Exercice

La suite de Padovan est définie de la manière suivante :

$$u_0 = u_1 = u_2 = 1 ;$$

$$u_{n+3} = u_{n+1} + u_n.$$

- a. Définir une constante `MAX_TAB` ayant pour valeur 20.
- b. Écrire en C/C++ une **fonction** `saisie_valeur` qui retournera au programme principal la valeur de `n` choisie par l'utilisateur. Attention, on recommencera la saisie **tant que la valeur n'est pas strictement positive et strictement inférieure à `MAX_TAB`.**
- c. Écrire en C/C++ une **procédure** `padovan` qui calcule et "renvoie" la valeur du  $n^{\text{ième}}$  terme de la suite.
- d. Écrire en C/C++ la procédure `remplit_tab` permettant de remplir le tableau avec les `n` premières valeurs de la suite, `n` étant passé en paramètre.
- e. Écrire en C/C++ la procédure `affiche_tab` permettant l'affichage du tableau obtenu ; attention, on n'affichera que les valeurs significatives du tableau (i.e. celles pour lesquelles le tableau a été rempli).
- f. Écrire en C/C++ le programme principal permettant de saisir une valeur `lambda`, et d'afficher les `lambda` premières valeurs de la suite en utilisant les sous-programmes précédents.

# Licence STS Université Claude Bernard Lyon1

## LIFAP1 – Séquence 3 – Groupes B, et D

### Sujet B

Contrôle Continu – TP6 - durée 40 min

Mardi 20 novembre 2018

### Consignes (À lire impérativement avant de commencer !!!)

Vous devrez compiler et tester votre programme à chacune des étapes de son élaboration.

Vous donnerez à votre fichier source le nom suivant : TP6.cpp

Dans votre fichier, **vous mettrez en commentaire vos nom et prénom ainsi que votre numéro d'étudiant et le sujet (A ou B).**

Vous devrez obligatoirement utiliser les notions vues en TD / TP durant le semestre et rien d'autre !!!

La note tiendra compte :

- du respect des consignes ;
- de la qualité de la présentation et de la lisibilité du code ;
- des algorithmes ;
- du bon fonctionnement du programme.

Une fois le programme terminé et testé (ou à la fin du temps imparti), vous devrez déposer le fichier source (.cpp) via **TOMUSS** (en cliquant sur "déposer" dans la case Depot\_TP6 de l'UE LIFAP1).

### Exercice

Soit un tableau 2D ayant au maximum MAX\_L lignes et MAX\_C colonnes et contenant des caractères.

'a'	'b'	'f'	'b'	'f'	'a'	'A'	'p'		
'#'	'a'	'v'	'Q'	'2'	'.'	'0'	'o'		
'4'	'o'	'['	'J'	'f'	'!'	'f'	'P'		
'è'	'v'	'K'	'a'	M	'Z'	'a'	'-'		
's'	'q'	'('	's'	'f'	'a'	'c'	'M'		

- 1- Définir, en langage C/C++, deux constantes MAX\_L et MAX\_C ayant pour valeurs respectives 8 et 10.
- 2- Ecrire, en langage C/C++, un sous-programme remplir\_tab, qui remplit les taille\_l premières lignes et taille\_c premières colonnes d'un tableau 2D de caractères en demandant les informations à l'utilisateur.
- 3- Ecrire, en langage C/C++, un sous-programme affiche\_tab, qui affiche les taille\_l premières lignes et taille\_c premières colonnes d'un tableau 2D de caractères.
- 4- Ecrire, en langage C/C++, une **procédure** compte\_occ, qui compte le nombre d'occurrence d'un caractère c passé en paramètre dans le tableau 2D.  
Exemple : compte\_occ (Tab, 'a', ...) → 6
- 5- Ecrire le programme principal qui demande à l'utilisateur le nombre réel de lignes (taille\_l) et de colonnes (taille\_c) à utiliser dans le tableau, le remplit, l'affiche, puis demande à l'utilisateur un caractère et compte le nombre d'occurrences de ce caractère dans le tableau.

# Licence STS Université Claude Bernard Lyon1

## LIFAP1 – Séquence 3 – Groupes B, et D

### Sujet A

Contrôle Continu – TP6 - durée 40 min

Mardi 20 novembre 2018

### Consignes (À lire impérativement avant de commencer !!!)

Vous devrez compiler et tester votre programme à chacune des étapes de son élaboration.

Vous donnerez à votre fichier source le nom suivant : TP6.cpp

Dans votre fichier, **vous mettrez en commentaire vos nom et prénom ainsi que votre numéro d'étudiant et le sujet (A ou B).**

Vous devrez obligatoirement utiliser les notions vues en TD / TP durant le semestre et rien d'autre !!!

La note tiendra compte :

- du respect des consignes ;
- de la qualité de la présentation et de la lisibilité du code ;
- des algorithmes ;
- du bon fonctionnement du programme.

Une fois le programme terminé et testé (ou à la fin du temps imparti), vous devrez déposer le fichier source (.cpp) via **TOMUSS** (en cliquant sur "déposer" dans la case Depot\_TP6 de l'UE LIFAP1).

### Exercice

Soit un tableau 2D ayant au maximum MAX\_L lignes et MAX\_C colonnes et contenant des caractères.

'a'	'b'	'f'	'b'	'f'	'a'	'A'	'p'	
'#'	'a'	'v'	'Q'	'2'	'.'	'0'	'o'	
'4'	'o'	'['	'J'	'f'	'!'	'f'	'P'	
'+'	'v'	'K'	'a'	M	'Z'	'a'	'-'	
's'	'q'	'('	's'	'f'	'a'	'c'	'M'	

- 1- Définir, en langage C/C++, deux constantes MAX\_L et MAX\_C ayant pour valeurs respectives 7 et 9.
- 2- Ecrire, en langage C/C++, un sous-programme `remplir_tab`, qui remplit les `taille_l` premières lignes et `taille_c` premières colonnes d'un tableau 2D de caractères en demandant les informations à l'utilisateur.
- 3- Ecrire, en langage C/C++, un sous-programme `affiche_tab`, qui affiche les `taille_l` premières lignes et `taille_c` premières colonnes d'un tableau 2D de caractères.
- 4- Ecrire, en langage C/C++, une **procédure** `compte_min_maj`, qui compte et "retourne" le nombre de caractères minuscule et le nombre de caractères majuscules dans le tableau 2D (dans l'exemple 22 minuscules et 8 majuscules).
- 5- Ecrire le programme principal qui demande à l'utilisateur le nombre réel de lignes (`taille_l`) et de colonnes (`taille_c`) à utiliser dans le tableau, le remplit, l'affiche, puis affiche le nombre de minuscules et de majuscules dans le tableau de taille `taille_l * taille_c`.

**Licence STS Université Claude Bernard Lyon1**  
**LIFAP1 – Séquence 3 – Groupes A, et C**  
**Sujet A**

Contrôle Continu – TP6 - durée 40 min  
Mardi 20 novembre 2018

**Consignes (À lire impérativement avant de commencer !!!)**

Vous devrez compiler et tester votre programme à chacune des étapes de son élaboration.

Vous donnerez à votre fichier source le nom suivant : TP6.cpp

Dans votre fichier, **vous mettrez en commentaire vos nom et prénom ainsi que votre numéro d'étudiant et le sujet (A ou B).**

Vous devrez obligatoirement utiliser les notions vues en TD / TP durant le semestre et rien d'autre !!!

La note tiendra compte :

- du respect des consignes ;
- de la qualité de la présentation et de la lisibilité du code ;
- des algorithmes ;
- du bon fonctionnement du programme.

Une fois le programme terminé et testé (ou à la fin du temps imparti), vous devrez déposer le fichier source (.cpp) via **TOMUSS** (en cliquant sur "déposer" dans la case Depot\_TP6 de l'UE LIFAP1).

**Exercice**

- 1- Ecrire en langage C/C++ une **fonction** `saisie_positive` permettant de saisir et de retourner une valeur choisie par l'utilisateur strictement positive. La fonction devra recommencer la saisie jusqu'à ce que la valeur soit strictement positive.
- 2- Ecrire en langage C/C++ une **fonction** booléenne `est_diviseur` qui renverra vrai si b divise a et faux sinon.  
Exemples : `est_diviseur (6,3)` renverra vrai et `est_diviseur (5,2)` renverra faux.
- 3- Ecrire en langage C/C++ une **procédure** `diviseurs_2_3_5` permettant de tester si un entier strictement positif passé en paramètre est divisible par 2, par 3, par 5. Cette procédure renverra 3 résultats booléens `div2`, `div3` et `div5`.  
Par exemple : 10 est divisible par 2 et 5 donc `div2 = true`, `div3=false` et `div5=true`.
- 4- Ecrire en langage C/C++ le programme principal permettant de saisir un nombre strictement positif et d'afficher s'il est divisible par 2, par 3, et/ou par 5.

**Licence STS Université Claude Bernard Lyon1**  
**LIFAP1 – Séquence 3 – Groupes A, et C**  
**Sujet B**

Contrôle Continu – TP6 - durée 40 min  
Mardi 20 novembre 2018

**Consignes (À lire impérativement avant de commencer !!!)**

Vous devrez compiler et tester votre programme à chacune des étapes de son élaboration.

Vous donnerez à votre fichier source le nom suivant : TP6.cpp

Dans votre fichier, **vous mettrez en commentaire vos nom et prénom ainsi que votre numéro d'étudiant et le sujet (A ou B).**

Vous devrez obligatoirement utiliser les notions vues en TD / TP durant le semestre et rien d'autre !!!

La note tiendra compte :

- du respect des consignes ;
- de la qualité de la présentation et de la lisibilité du code ;
- des algorithmes ;
- du bon fonctionnement du programme.

Une fois le programme terminé et testé (ou à la fin du temps imparti), vous devrez déposer le fichier source (.cpp) via **TOMUSS** (en cliquant sur "déposer" dans la case Depot\_TP6 de l'UE LIFAP1).

## **Exercice**

Un nombre heureux est un nombre entier positif qui, lorsqu'on ajoute les carrés de chacun de ses chiffres, puis les carrés des chiffres de ce résultat et ainsi de suite **jusqu'à l'obtention d'un nombre à un seul chiffre**, donne 1 pour résultat. Exemple 7 est heureux, puisque :  $7^2 = 49$  et  $4^2 + 9^2 = 97$  et  $9^2 + 7^2 = 130$  et  $1^2 + 3^2 + 0^2 = 10$  et  $1^2 + 0^2 = 1$  (on est arrivé à un nombre d'un seul chiffre = 1, donc 7 est heureux).

1. Écrire l'algorithme d'une **procédure** `somme_carre` permettant de calculer la somme des carrés des chiffres composant un nombre entier passé en paramètres. Exemple `somme_carre(49) → 97`
2. Écrire l'algorithme d'une fonction booléenne `heureux?` qui retourne vrai si un entier `n` passé en paramètre est heureux, faux sinon
3. Écrire le programme principal permettant d'afficher tous les nombres heureux inférieurs à 100