



# CENTRE SCOLAIRE SAINTE-JULIENNE

## PL -TE 0 – Projet Da Love Machine

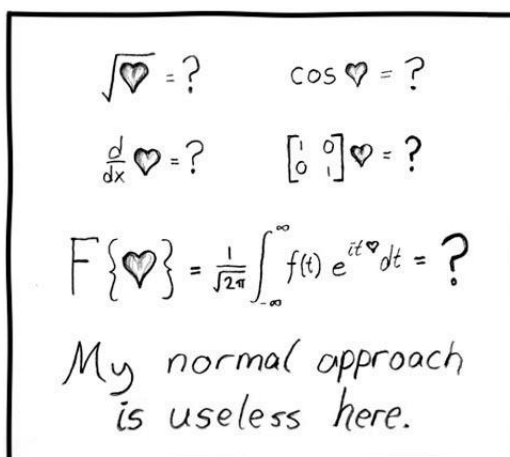
### Consignes

#### Invariant

Développer une application non orientée objet sur la base d'un cahier des charges intégrant des fonctions personnalisées, des structures imbriquées et un tableau à plusieurs dimensions.

#### Mise en situation

Tu es un dragueur dans l'âme et tu cherches à démontrer à ta copine que vous êtes le couple idéal. Dans le cadre de ce projet tu vas implémenter un algorithme de numérologie: da love machine. L'idée générale est de présenter le pourcentage de compatibilité de deux personnes à partir de leur prénom.



Vous croyez avoir trouvé votre âme sœur, mais voulez-vous en avoir le cœur net? Vous ne serez pas déçu d'être venus aujourd'hui, car vous allez apprendre le secret d'un couple heureux grâce à la numérologie des prénoms.

« Parce que vos prénoms en disent long sur vos personnalités, découvrez s'ils sont faits l'un pour l'autre. »

L'algorithme infallible pour calculer la compatibilité de deux prénoms que vous allez implémenter est le suivant.

Etant donné deux prénoms (tristan et yseult), **on remplace chaque lettre par un chiffre entre 1 et 9**, qui correspond au tableau suivant :

1	2	3	4	5	6	7	8	9
a	b	c	d	e	f	g	h	i
j	k	l	m	n	o	p	q	r
s	t	u	v	w	x	y	z	

Vous remarquerez donc que la valeur d'une lettre est sa position dans l'alphabet (en commençant à 0!), modulo 9, plus 1. Par exemple, k est la lettre 10 de l'alphabet,  $10 \% 9 + 1 = 2$ . a est la lettre 0 de l'alphabet,  $0 \% 9 + 1 = 1$ . Il est possible de trouver la position à partir du code ASCII - 97.

tristan → [2,9,9,1,2,1,5]  
yseult → [7,1,5,3,3,2]

On **additionne ensuite les chiffres du prénom deux par deux**, en laissant le dernier intact s'il y en a un nombre impair:

tristan → [11,10,3,5]  
yseult → [8,8,5]

On **transforme chaque nombre au-dessus de 9 en un chiffre de 1 à 9** en additionnant les chiffres qui le composent. Par exemple, 11 donnera 2.

tristan → [2,1,3,5]  
yseult → [8,8,5]

On additionne deux par deux les chiffres qui restent:

tristan → [3,8]  
yseult → [16,5]

On transforme à nouveau les nombres supérieurs à 9 en chiffres de 1 à 9:

tristan → [3,8]  
yseult → [7,5]

On additionne à nouveau deux par deux les chiffres:

tristan → [11]  
yseult → [12]

On transforme à nouveau les nombres supérieurs à 9 en chiffres:

tristan → [2]  
yseult → [3]

Quand on a plus qu'un seul chiffre (le **score**) pour chaque prénom, on calcule la compatibilité entre les deux. Plus la différence entre eux est grande, moins il y a de compatibilité:

Différence	0	1	2	3	4	5	6	7	8
Compatibilité	100%	87.5%	75%	62,5%	50%	37,5%	25%	12.5%	0%

On peut donc dire avec certitude que la compatibilité amoureuse entre Tristan et Yseult est de 87.5%.

Maintenant, implémentez et testez cet algorithme: entrez votre prénom et celui de quelqu'un d'autre, et voyez si ça vaut le coup d'essayer ou si ... next!

## Les objets d'apprentissage

Appliquer	Transférer
<ul style="list-style-type: none"><li>• Lire un algorithme intégrant des structures imbriquées.</li><li>• Traduire un algorithme intégrant des structures imbriquées dans un langage de programmation, en respectant la syntaxe.</li><li>• Programmer en utilisant un tableau à plusieurs dimensions.</li><li>• Programmer en recourant à des fonctions personnalisées.</li><li>• Commenter les lignes de codes.</li><li>• Tester le programme conçu.</li></ul>	<ul style="list-style-type: none"><li>• Extraire d'un cahier des charges les informations nécessaires à la programmation.</li><li>• Écrire un algorithme intégrant des structures imbriquées.</li><li>• Programmer en recourant aux fonctions personnalisées, instructions et types de données nécessaires au développement d'une application.</li><li>• Corriger un programme défaillant.</li><li>• Améliorer un programme pour répondre à un besoin défini.</li></ul>
Connaître	
<ul style="list-style-type: none"><li>• Expliquer la notion de portée d'une variable.</li><li>• Différencier un langage interprété d'un langage compilé.</li><li>• Expliquer la structure d'un tableau.</li><li>• Expliquer la syntaxe d'utilisation des fonctions principales d'un tableau, dont le calcul de la taille, l'insertion et la suppression d'un élément.</li><li>• Associer les fonctions principales d'un tableau à leur utilisation.</li><li>• Expliquer la syntaxe d'utilisation de fonctions personnalisées.</li></ul>	

## Les exigences techniques

1. Il faut mettre en place des règles de vérification assurant une saisie correcte des prénoms;
2. Les manipulations sur les prénoms doivent s'effectuer exclusivement sur des minuscules;
3. Les prénoms sont stockés et utilisés avec des tableaux dynamiques et/ou des listes;
4. Le code utilise des fonctions à but spécifique;
5. L'utilisation de la récursivité entrainera un bonus dans l'évaluation.

## **L'organisation du projet**

Il s'opère en groupe de 2 lors des labos et à la maison. Les points de l'évaluation sont alloués à un rapport écrit.

Le rapport écrit doit reprendre les éléments suivants:

- Les algorithmes exprimés sous forme de pseudocodes;
- Le code source de votre application;
- Des captures d'écran illustrant divers scénarios.

Ce projet représente 50% des points de l'examen Juin et devra être remis une semaine avant le début de la session d'examen de Juin. Je vous signale que le respect des échéances sera évalué.

## **Les critères d'évaluation**

Ils sont repris sur une grille d'évaluation.

Vous serez évalués sur la compétence suivante:

UAA11    Programmation procédurale    Développer une application non orientée objet sur la base d'un cahier des charges intégrant des fonctions personnalisées, des structures imbriquées et un tableau à plusieurs dimensions

Je vous souhaite un bon travail!