

# (M1102) Test d'algorithmique

Le 17 Octobre 2015 – durée : 1 heure 20 minutes

© Alain Casali alain.casali@univ-amu.fr

I.U.T.d'Aix en Provence - Département Informatique



Documents autorisés : une feuille simple A4 manuscrite recto / verso.

Remarques

- Lire attentivement tout le sujet, avant de commencer. Le barème, donné à titre indicatif, est sur 20 points, et 3 de bonus ;
- Tout au long de ce test, donnez de la sémantique à vos variables ! Toute variable n'ayant pas la sémantique adaptée sera considérée comme fausse, et par conséquent le morceau de code dans laquelle elle est présente aussi ;
- Toutes les fonctions écrites ou utilisées en cours ou en TD sont utilisables dans l'état ;
- Écriture au crayon de papier interdite.

Le but de ce test est d'écrire quelques sous-programmes permettant de crypter ou de décrypter une chaîne de caractères.

On rappelle que les fonctions `succ ()` et `pred ()` renvoient respectivement le caractère suivant, respectivement le caractère précédent, selon l'encodage ASCII. Le profil de ces fonctions est :

```
fonction succ (Car : in caractere) renvoie caractere ;
fonction pred (Car : in caractere) renvoie caractere ;
```

## Exercice 1

Dans cet exercice, on s'intéresse à la méthode de cryptage dite de *César*. Elle est aussi connue sous le nom de «*chiffrement par décalage*». Cette méthode consiste simplement à décaler les lettres de l'alphabet de quelques crans vers la droite ou la gauche. Pour cet exercice, on considère le décalage d'une unité vers la droite.

Question 1.1 (1 point) :

Soit la fonction `DecrypteCesar ()` de profil :

```
fonction DecrypteCesar (Str : in string) renvoie string ;
```

Transformer le profil de cette fonction en une procédure.

Question 1.2 (2 points) :

Les romains n'ayant pas accès à la table des caractères ASCII, il faut redéfinir les fonctions `succ ()` et `pred ()` de façon à toujours rester dans l'alphabet. En conséquence, le suivant du caractère 'Z' est le caractère 'A', celui de 'z' est 'a'. Dans les autres cas, la fonction `succ ()` renvoie la bonne valeur.

Écrire le corps de la fonction `MySucc ()` de profil :

```
fonction MySucc (C : in caractere) renvoie caractere ;
```

selon les spécifications décrites ci-dessus.

NB : on dispose aussi de la fonction `MyPred ()`. Cette fonction a le même profil que `MySucc ()`, est fait le travail inverse.

Question 1.3 (2 points) :

Puisque les romains n'ont pas accès à la table ASCII, ce qui n'est pas notre cas, il faut pourvoir tester si un caractère est un caractère alphabétique. Pour cela, vous devez écrire le corps du prédicat `IsAlpha ()` qui renvoie vrai si le caractère passé en paramètre est un caractère de l'alphabet (majuscule ou minuscule), faux sinon. Le profil de ce prédicat est le suivant :

```
fonction IsAlpha (C : in caractere) renvoie boolean ;
```

Question 1.4 (3 points) :

Écrire le corps de la fonction `DecrypteCesar ()` selon les spécifications de l'énoncé.

Question 1.5 ( $\frac{1}{2}$  point bonus) :

Quel est le résultat du décryptage pour la chaîne «Vo hffl of dsjf qbt, jm VSM» ?

Question 1.6 (4 points) :

Écrire l'algorithme suivant qui :

- entre dans une boucle infinie ;
- demande à l'utilisateur de saisir une chaîne ;
- sort de la boucle infinie si la taille de la chaîne est nulle ;
- encode la chaîne saisie à l'aide de la fonction `CrypteCesar ()` de profil :  

```
fonction CrypteCesar (Str : in string) renvoie string;
```
- affiche la chaîne cryptée ;
- compare le résultat du décryptage à la chaîne initiale :  
— s'il y a une différence, on sort de la boucle infinie.

---

## Exercice 2

Le but de cet exercice est d'écrire d'autres fonction d'encodage d'une chaîne de caractères selon les spécifications propres à chaque question.

Question 2.1 (2 points) :

Le décalage à droite d'un unique élément étant trop simple, on peut encoder la chaîne en décalant chaque caractère de `Cle` éléments. Si `Cle` vaut 3 ; alors on obtient le cryptage utilisé par César. À titre d'exemple, voici l'alphabet crypté utilisé :

Lettre	a	b	c	...	z
Lettre Cryptée	d	e	f	...	c

Écrire le corps de la fonction `CrypteCesar2 ()` selon les spécifications de l'énoncé. Le profil de cette fonction est le suivant :

```
fonction CrypteCesar2 (Str : in string, Cle : in entier_naturel) renvoie string;
```

Question 2.2 (3 points) :

Pour cette question, les spécifications du cryptage sont les suivantes :

- on a accès l'intégralité de la table d'encodage car on ne souhaite plus utiliser le prédicat `IsAlpha ()`. En conséquence, on va encoder tous les caractères de la chaîne ;
- chaque caractère de la chaîne est décalé sur la droite autant de fois que son indice (sa position) plus un dans la chaîne. En conséquence, le premier est décalé d'une unité, le second de deux, le troisième de trois, etc ...

Écrire le corps de la fonction `CrypteCesar3 ()` selon les spécifications de l'énoncé. Le profil de cette fonction est le suivant :

```
fonction CrypteCesar3 (Str : in string) renvoie string;
```

Question 2.3 (3 points) :

Pour cette question, les spécifications du cryptage sont les suivantes :

- on a accès l'intégralité de la table d'encodage car on ne souhaite plus utiliser le prédicat `IsAlpha ()`. En conséquence, on va encoder tous les caractères de la chaîne ;
- si le caractère est d'indice pair, on fait un décalage sur la droite, sinon on fait un décalage sur la gauche.

Écrire le corps de la fonction `CrypteCesar4 ()` selon les spécifications de l'énoncé. Le profil de cette fonction est le suivant :

```
fonction CrypteCesar4 (Str : in string) renvoie string;
```

Question 2.4 ( $2\frac{1}{2}$  points bonus) :

Écrire le corps de la fonction `CrypteCesar5 ()` de profil :

```
fonction CrypteCesar5 (Str : in string) renvoie string;
```

Pour cette fonction, vous devez reprendre les spécifications des deux dernières questions, à savoir :

- on a accès l'intégralité de la table d'encodage car on ne souhaite plus utiliser le prédicat `IsAlpha ()`. En conséquence, on va encoder tous les caractères de la chaîne ;

- si le caractère est d'indice pair, on fait un décalage sur la droite, sinon on fait un décalage sur la gauche ;
  - chaque caractère de la chaîne est décalé sur la droite (ou sur la gauche) autant de fois que son indice (sa position) plus un dans la chaîne.
-