***SIO CCF : Le jeu du Pendu***

*A rendre pour le dimanche 29 décembre dernier délai par envoi sur l’ENT ou dans le casier*

**Partie A - Durée : 30 mn**

**Les réponses sont à rédiger sur une copie**

|  |
| --- |
| **A-1** Écrire un algorithme chercheL prenant pour entrées deux chaînes de caractères m et k, et renvoyant un booléen indiquant si la lettre k est présente dans le mot m. (Vrai si lettre présente, Faux si lettre non présente) |

#Algo\_chercheL

#Variables

Ecrire(« Entrez une chaînes de caractères »)

Lire(m) : <chaine de caractères>

Entrer(« Entrez une lettre »)

Lire(k) : <chaine de caractères>

L ← long(m) : <entier>

reponse = false : <booléen>

#Traitement

Pour i allant de 0 à L-1:

Si m[i] = k :

reponse ← true

FinSi

FinPour

Afficher(reponse)

#Fin

**A-2** On considère l'algorithme mystere ci-dessous.

a) Quelle est la valeur retournée par mystere si l’utilisateur entre x="lependu" et c="c" ?

La valeur retournée par mystere sera : 0

b) Quelle est la valeur retournée par mystere si l’utilisateur entre x="lependu" et c="e" ?

La valeur retournée par mystere sera : 2

c) Quel est le rôle de cet algorithme ?

Le rôle de cet algorithme est de compter le nombre de lettre existant dans une chaine de caractères en fonction d’un caractère donné.

A-3 Écrire l'algorithme d'un programme principal permettant :

la saisie par le joueur 1 d'un mot « secret » ;

• les saisies successives par le joueur 2 de lettres proposées

(au maximum 10 saisies, et on arrête avant si le joueur 2 gagne);

• l'affichage après chaque lettre proposée du type

« La lettre fait partie du mot, il reste … lettres à deviner.»

(ou « … ne fait pas partie ... ») ;

• l'affichage final du type « Le joueur ... a gagné ».

*Remarque : on suppose dans cette question que le joueur 2 ne propose que des lettres différentes.*

#Algo\_programme\_principal

#Variables

Ecrire(« Le joueur 1 rentre un mot secret »)

Lire(secret) : <chaine de caractère>

essai ← 1 : <entier>

r ← False : <booléen>

trouve ← 0 : <booléen>

reste ← long(secret) : <booléen>

#Traitement

Tant que n <= 10 et reste != 0 :

Entrer(« Le joueur 2 rentre une lettre\n »)

Lire(lettre) : <string>

Pour i allant de 0 à long(secret) :

Si secret[i] = lettre :

R ← True

trouve ← trouve+1

FinSi

Sinon :

R ← False

FinSinon

FinPour

reste ← long(secret) – trouve

Si r = True :

Ecrire(« La lettre fait partie du mot, il reste», reste, « dans le mot » )

FinSi

Sinon

Ecrire(« La lettre ne fait pas partie du mot, il reste», reste,

« dans le mot » )

FinSinon

Si reste = 0 :

Ecrire(« Le joueur 2 a gagné »)

FinSi

Sinon

Ecrire(« Le joueur 1 a gagné»)

FinSinon

#Fin

**A-4** Ecrire l'algorithme un algorithme positionL prenant pour entrées deux chaînes de caractères m et k, et renvoyant une chaîne indiquant les positions de la lettre k dans le mot m.

*Exemple :* Pour m="lependu" et k="c", positionL retourne "" et pour m="lependu" et k="e", positionL retourne "2;4"

Expliquer alors comment modifier le programme principal de façon à ce qu'il affiche en plus les positions de la lettre proposée.

*Exemple :« La lettre est dans le mot aux positions 2;4, il reste … lettres à deviner.» (ou « … ne fait pas partie ... »)*

#Algo\_positionL

#Variables

Entrer(« Entrez une chaînes de caractères »)

Lire(m) : <string>

Entrer(« Entrez une lettre »)

Lire(k) : <string>

L ← long(m) : <int>

reponse ← « » : <string>

#Traitement

Pour i allant de 0 à L :

Si m[i] = k :

reponse ← reponse+string(i+1)+ « ; »

FinSi

FinPour

Retourner(reponse)

#Fin

Pour modifier le programme, nous avons juste à ajouter une variable « position » prenant la valeur  « chaine vide » et faire cela :

Si secret[i] = lettre :

R ← True

trouve ← trouve+1

position ← position+string(i+1)+ « ; »

FinSi

Puis :

Si r = True :

Ecrire(« La lettre fait partie du mot, il reste», reste, « dans le mot » )

Ecrire(« La lettre se trouve à la position suivante : »,position)

FinSi