

b) Construire alors un autre programme permettant de calculer les 10 premiers termes de la somme précédente et affichant la valeur approchée de π obtenue.

c) La formule de Wallis peut se transformer successivement en :

$$\begin{aligned}\frac{\pi}{2} &= 1 + \frac{1}{3} + \frac{1 \times 2}{3 \times 5} + \frac{1 \times 2 \times 3}{3 \times 5 \times 7} + \frac{1 \times 2 \times 3 \times 4}{3 \times 5 \times 7 \times 9} + \dots \\ &= 1 + \frac{1}{3} \left(1 + \frac{2}{5} \left(1 + \frac{3}{7} \left(1 + \frac{4}{9} (1 + \dots) \right) \right) \right)\end{aligned}$$

dont une approximation est, pour un entier k donné :

$$\frac{\pi}{2} \approx 1 + \frac{1}{3} \left(1 + \frac{2}{5} \left(1 + \frac{3}{7} \left(1 + \frac{4}{9} \left(\dots \left(1 + \frac{k}{2k+1} \right) \right) \right) \right) \right)$$

1) Lorsque $k = 4$, simuler, sans utiliser la calculatrice, le calcul.

2) A partir d'un entier k donné au clavier, calculer la valeur approchée de π obtenue.

(Vous commencerez votre boucle de calcul par $1 + \frac{k}{2k+1}$)

Exercice 4 (4,5 points)

Dans l'ensemble des nombres entiers, il existe des "trous" entre deux nombres premiers dont la longueur dépasse tout nombre fixé à l'avance. Par exemple, l'intervalle $[23;29]$ est un trou de longueur 6 (il n'existe aucun nombre premier entre 23 et 29), $[89;97]$ est un trou de longueur 8 et $[113;127]$ est un trou de longueur 14.

Le but de cet exercice est de faire afficher un trou de longueur au moins égale à un entier entré au clavier par l'utilisateur.

Vous devrez choisir l'une des deux méthodes suivantes ; la première utilise les fonctions, la seconde non.

Version 1

1) Ecrire en Scilab une fonction **[b]=premier(n)** retournant un booléen b indiquant si l'entier n reçu en paramètre est premier ou non.

2) Utiliser la fonction précédente dans un programme Scilab permettant de résoudre le problème posé. Vous donnerez une description en français de la solution retenue.

Version 2

1) Ecrire en Scilab un programme affichant VRAI ou FAUX suivant si un entier n donné au clavier est premier ou non.

2) Utiliser les instructions du programme précédent dans un programme Scilab permettant de résoudre le problème posé. Vous donnerez une description en français de la solution retenue.

Vous devrez rendre 3 copies :

- Une copie pour l'exercice 1**
- Une copie pour l'exercice 2**
- Une copie pour l'exercice 3**

Une partie du barème est donnée sur la syntaxe du langage Scilab et sur votre interprétation de l'algorithme nécessaire pour répondre à chaque problème. Vous devrez donc expliquer chaque programme par un texte en français.

Exercice 1 (6 points)

Critère de divisibilité par 11 :

Pour qu'un nombre soit divisible par 11, il faut et il suffit que la différence entre la somme des chiffres de rang impair et la somme des chiffres de rang pair, à partir de la droite, soit égale à 0 ou divisible par 11. Par exemple, 18282 est divisible par 11 car $(2 + 2 + 1) - (8 + 8) = 5 - 16 = -11$ qui est divisible par 11. En fait $18282 = 11 \times 1662$.

Ecrire un programme Scilab permettant de déterminer si un nombre n à 5 chiffres lu au clavier est divisible ou non par 11 (par le critère expliqué ci-dessus).

Vous commencerez votre programme par une décomposition du nombre en ses chiffres.

Exercice 2 (8 points)

Problème du voyageur

Un voyageur doit se rendre dans plusieurs villes et souhaite trouver un parcours avantageux, permettant de passer dans toutes les villes en partant d'une ville de départ donnée. Les distances, deux à deux, entre les villes sont connues.

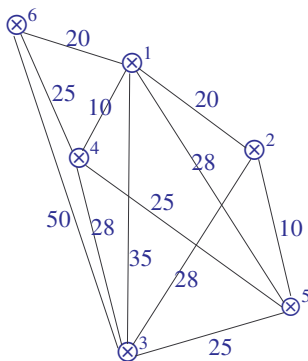
Nous supposons qu'il utilise la stratégie suivante :

A partir de chaque ville, il choisit de se rendre à la ville la plus proche qu'il n'ait pas encore visitée.

L'ensemble des distances peut être donné sous forme d'un tableau carré symétrique ayant la diagonale principale constituée de 0 (la distance d'une ville à elle-même est nulle).

Exemple :

Soit le graphe suivant, représentant les connexions entre les six villes :



Le tableau des distances de l'exemple est :

	V1	V2	V3	V4	V5	V6
V1	0	20	35	10	28	20
V2	20	0	28	20	10	35
V3	35	28	0	25	25	50
V4	10	20	25	0	25	25
V5	28	10	25	25	0	45
V6	20	35	50	25	45	0

Les données seront connues dans un tableau à deux dimensions.

Il faut donc, en partant de la ville où l'on se trouve, rechercher le minimum de la ligne (si l'on fait la lecture par ligne) correspondante, en excluant les villes déjà visitées. Pour cela, il suffit de remplacer à chaque étape, toutes les distances d'une colonne correspondant à la ville où l'on se trouve, par la plus grande des distances possible, par exemple par le nombre entier 1000 (nous

supposerons ainsi que les villes ne sont pas distantes entre elles d'une distance supérieure à 1000 km).

- 1) Ecrire la fonction **[t]=Saisir** demandant à l'utilisateur d'initialiser le tableau **t** des distances par une lecture au clavier.
- 2) a) Soit **tab** un tableau à une dimension. Construire la fonction **[n]=minimum(tab)** donnant l'indice de l'élément le plus petit de ce tableau **tab**.
b) Modifier la fonction précédente de manière à écrire la fonction **[j]=min_ligne(L,t)** donnant l'indice de colonne **j** du plus petit élément de la ligne **L** du tableau **t**.
- 3) Ecrire la fonction **[tbis]=déjà_fait(c,t)** remplaçant tous les éléments de la colonne repérée par l'entier **c** du tableau **t** par 1000 et construisant le tableau **tbis** ainsi modifié.
- 4) Ecrire un programme Scilab utilisant les fonctions précédentes et partant de la ville numérotée 1. Vous indiquerez la distance parcourue et afficherez les étapes successives comme, en prenant la solution de l'exemple (ou tout autre mode d'affichage) :

```
VILLE : 1  DISTANCE : 0
VILLE : 4  DISTANCE : 10
VILLE : 2  DISTANCE : 30
VILLE : 5  DISTANCE : 40
VILLE : 3  DISTANCE : 65
VILLE : 6  DISTANCE : 115
```

Exercice 3 (7 points)

Un générateur de Charabia latin

Nous nous proposons de réaliser un générateur de charabia latin (jeu utilisé par les petits enfants de langue anglaise), c'est-à-dire un programme qui modifie un mot du français en un mot d'un charabia latin. Cette transformation s'effectue en plaçant la première lettre du mot à la fin, et en y ajoutant la lettre "a". Ainsi, le mot "tortue" devient "ortueta", "Scilab" devient "cilaba", et ainsi de suite.

Ecrivons maintenant un programme Scilab qui lira une phrase en français et écrira son équivalent en charabia latin. Pour des raisons de simplicité, nous ne tiendrons pas compte du problème des lettres majuscules et des signes de ponctuation.

L'écriture du programme traduisant le fonctionnement de ce jeu sera facilitée par la construction des modules suivants :

- 1) Ecrire la fonction **[ch]=Entrez_Texte** permettant d'entrer le texte en français représenté par la chaîne de caractère **ch** (vérifiez que le texte comporte moins de 80 caractères).
- 2) Ecrire la fonction **[n]=Nombre_mots** permettant de déterminer le nombre **n** de mots de la phrase **ch** en utilisant le nombre d'espaces.
- 3) Ecrire la fonction **[m]=recherche_mot(ch,n)** permettant d'affecter à la chaîne **m** le **n**^{ième} mot de la phrase **ch**.
- 4) Ecrire la fonction **[ma]=Transf_Franc_Charabia(mf)** permettant de transformer le mot français de la chaîne **mf** en son équivalent en charabia latin dans la chaîne **ma**.
- 5) Ecrire le programme complet permettant de transformer une phrase complète en son équivalent en charabia latin.

**Vous devrez rendre 3 copies : Une copie pour l'exercice 1
Une copie pour l'exercice 2
Une copie pour l'exercice 3**

Une partie du barème est donnée sur la syntaxe du langage Scilab et sur votre interprétation de l'algorithme nécessaire pour répondre à chaque problème. Vous devrez donc expliquer chaque programme par un texte en français.

Exercice 1 (6 points)

Modélisation d'un échange entre deux milieux

Deux récipients A et B sont séparés par une membrane perméable dans les deux sens. On place dans les récipients A et B deux solutions contenant respectivement a_0 molécules (dans A) et b_0 molécules (dans B). On suppose que, toutes les heures, 20% des molécules passent de A dans B et 10% des molécules passent de B dans A . On note a_n et b_n les nombres respectifs de molécules présentes dans A et B au bout de n heures.

1) Montrer que $\begin{cases} a_{n+1} = 0,8a_n + 0,1b_n \\ b_{n+1} = 0,2a_n + 0,9b_n \end{cases}$ et donner l'interprétation matricielle de ce système en considérant la matrice colonne $p_n = \begin{pmatrix} a_n \\ b_n \end{pmatrix}$.

Les deux récipients n'ayant d'échanges qu'entre eux.

2) Sachant que si $a_0 = 150$ et $b_0 = 20$ (unités), quelles instructions écrire pour connaître les quantités de molécules après 10 heures ?

Quelle méthode appliqueriez vous pour connaître la répartition limite, si elle existe, entre les deux milieux ?

3) Quels sont les dosages initiaux nécessaires pour obtenir après 1 heure, une répartition égale à $a_1 = 130$ et $b_1 = 40$ (unités). Ecrire l'instruction Scilab permettant d'explicitier le résultat.

Exercice 2 (6 points)

Soit l'algorithme suivant :

- (1) entrer au clavier un entier naturel
- (2) l'élever au carré
- (3) ajouter au résultat le nombre initialement choisi
- (4) à ce dernier résultat ajouter 41
- (5) afficher ce dernier résultat

1) Ecrire un programme en Scilab permettant de réaliser cet algorithme.

(Vous devrez vous assurer que l'entier qui est proposé par l'utilisateur est effectivement positif.)

2) Le programme suivant consiste à vérifier si, au départ, on choisit un nombre entier compris entre 0 et 39, alors on obtient toujours un nombre premier.

Vous devez construire le programme en respectant les étapes successives qui sont décrites ci-après :

a) construire une boucle en la variable i permettant de parcourir les entiers entre 0 et 39.

b) reprendre la partie du programme réalisé en 1) permettant d'affecter à la variable x le résultat de l'algorithme pour la valeur initiale i .

c) Tester si cet entier x est un nombre premier ou non.

(Un nombre est premier si, parmi tous les nombres entiers entre 1 et lui-même, il n'est divisible que par 1 et lui-même).

d) Vous compterez en la variable **somme** le nombre d'entiers donnant par cet algorithme un nombre premier. Suivant le résultat de cette variable, affichez si tous les nombres choisis entre 0 et 39 produisent un nombre premier.

Remarque : Cette propriété est vérifiée avec des nombres autres que 41. A l'instruction (4), en ajoutant un nombre Y au lieu de 41, on obtient des nombres premiers quand on choisit une valeur initiale comprise entre 0 et $Y - 2$.

Ces nombres Y sont appelés les nombres chanceux d'Euler (mathématicien suisse 1707-1783).

Il n'en existe que six : 2, 3, 5, 11, 17 et 41.

Exercice 3 (9 points)

Partie A

1) Construire la **fonction [tab]=nombre_mot(ch)** permettant de construire le tableau tab à partir de la chaîne de caractères ch de la façon suivante :

Le tableau tab va contenir le nombre de lettres de chaque mot. Les caractères séparateurs entre les mots seront à considérer parmi le *point*, la *virgule*, l'*apostrophe* et l'*espace*. Vous pourrez supposer que la phrase ne contient pas plus de 100 mots.

Exemple : A partir du texte du célèbre quatrain "Que j'aime à faire apprendre...", le tableau aura la forme suivante :

3	1	4	1	5	9	...
---	---	---	---	---	---	-----

2) Construire la **fonction [c]=nombre(tab)** permettant de transformer le tableau tab en la chaîne de caractère qui est la concaténation de tous les chiffres donnés par le tableau. Sur l'exemple précédent, cette fonction devra renvoyer la chaîne de caractères c = '314159' (et permettant ainsi de reconstituer les chiffres du nombre π).

3) Supposons que l'on dispose du tableau tab de la question de la partie A) 1). Il s'agit de compter le nombre d'occurrences de chaque chiffre (le nombre de 0, le nombre de 1, ...).

Construire la **fonction [tab2]=occurrence(tab)** permettant de compter le nombre de chaque chiffre du tableau tab (supposé de taille 100) pour construire le tableau tab2 de taille 10 (la position 1 pour le chiffre 1, 2 pour le chiffre 2, ..., 9 pour le chiffre 9 et 10 pour le chiffre 0).

Partie B

On souhaite connaître si le tableau d'occurrences trouvées précédemment peut être conforme à un modèle probabiliste (variable aléatoire). Nous allons pour cela utiliser le test dit du χ^2 (Khi-Deux).

Principe :

Ce test est basé sur le calcul des distances entre les valeurs observées et les valeurs théoriques. Il est évident que, plus cette valeur est importante, plus l'écart entre les valeurs observées et les valeurs théoriques est important et donc que le relevé statistique ne suit peut-être pas le modèle probabiliste donné. Pour mesurer

ces écarts, on considère la variable aléatoire $D^2 = \sum_{i=1}^k \frac{(n_i - np_i)^2}{np_i}$

où : k est le nombre d'éventualités de la variable aléatoire,
 n_i sont les effectifs observés,
 p_i est la probabilité d'apparition de la $i^{\text{ème}}$ éventualité
 n est l'effectif total des valeurs observées.

On montre que la loi de D^2 est proche de celle d'une variable unique notée χ^2 dont les déciles d_α sont donnés par la table ci-après dans le cas du neuvième décile. Lorsque la loi théorique est connue, $q = k - 1$ est appelé "nombre de degrés de liberté" de cette variable.

On convient, pour une expérience aléatoire, de rejeter la "régularité" de façon significative au seuil de 10% lorsque la valeur observée de D^2 dépasse d_9 (d_9 est le neuvième décile, c'est-à-dire la valeur telle que 90% de l'effectif total soit inférieur à cette valeur) :

Soit d^2 la valeur observée de D^2 , si $d^2 > d_9$, au seuil de 10%, on décide de rejeter l'hypothèse de régularité et donc que notre relevé statistique ne suit pas la loi théorique (au risque de se tromper de 10%).

q	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
d_9	2,71	4,61	6,25	7,78	9,24	10,64	12,02	13,36	14,68	15,99	17,27	18,55	19,81	21,06	22,31

Application :

1) Calculer le nombre $D^2 = \sum_{i=1}^k \frac{(n_i - np_i)^2}{np_i}$ donné dans le paragraphe 'Principe' de cet exercice en construisant

la **fonction [d]=distance(tab)** : nous aurons dans cet exemple, en supposant que les chiffres des décimales de π suivent la loi uniforme sur les entiers entre 0 et 9 et que les n_i sont les effectifs de chacun des chiffres :

$$D^2 = \sum_{i=0}^9 \frac{(n_i - 100 \frac{1}{10})^2}{100 \frac{1}{10}} = \sum_{i=0}^9 \frac{(n_i - 10)^2}{10} = \sum_{i=0}^9 \frac{(\text{effectifs observés} - \text{effectifs théoriques})^2}{\text{effectifs théoriques}}$$

2) Supposons que la table de la loi du χ^2 (donnée dans l'introduction) soit donnée dans un tableau appelé table. A partir de la valeur d trouvée dans la question 2), construire la **fonction seuil(d)** permettant d'afficher 'relevé conforme à la loi' ou 'relevé non-conforme à la loi'.

(Remarque : la valeur de q dite degré de liberté de la loi du χ^2 est ici égale à 9)