REPUBLIQUE TUNISIENNE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Concours Nationaux d'Entrée aux Cycles de Formation d'Ingénieurs
Session 2012

الجمهورية التونسية
وزارة التعليم العالي والبحث العلمي
المناظرات الوطنية للدخول إلى مراحل تكوين المهندسين
دورة 2012

Concours Toutes Options
Epreuve d'Informatique

Date : Mardi 05 Juin 2012	**Heure : 15 H**	**Durée : 2 H**	**Nbre pages : 6**

Barème : EXERCICE : 4 points
PROBLEME 1 : 6 points
PROBLEME 2 : 10 points

DOCUMENTS NON AUTORISES
L'USAGE DES CALCULATRICES EST INTERDIT

EXERCICE (Maple)

On considère la matrice carrée $A=\left(a_{i,j}\right)$ d'ordre 5 tel que :

$$a_{i,j}=\begin{cases} 1-\left(\dfrac{1}{2}\right)^{5-i} & \text{Si } i=j \text{ et } i<5 \\ 1 & \text{Si } i=5 \text{ et } j=5 \\ \left(\dfrac{1}{2}\right)^{5-j} & \text{Si } i=j+1 \\ 0 & \text{Sinon} \end{cases}$$

Donner les instructions Maple permettant de :

1. définir la fonction f à deux variables i et j tel que $f(i,j)=a_{i,j}$;
2. définir A, en utilisant la fonction f ;
3. calculer d, le déterminant de A ;
4. déterminer L, la liste des valeurs propres de A ;
5. déterminer AI, la matrice inverse de A ;
6. donner une réduction de Jordan de A puis afficher la matrice de passage ;
7. définir $I5$ la matrice identité d'ordre 5 ;
8. définir la fonction P en X tel que $P(X)=\det(X.I5-A)$;
9. vérifier l'égalité $P(A)=0$ sachant que $P(X)$ est le polynôme caractéristique de A ;
10. résoudre le système linéaire $A.x=b$ avec $b=\begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix}$.

PROBLEME 1 (Maple)

PARTIE I

Le nombre de changements de signe, entre les éléments d'une liste $[y_0, y_1, y_2, \ldots, y_n]$ de $n+1$ réels non nuls, est égal au nombre d'éléments de l'ensemble $E = \{k \in [1,n] \text{ tel que } y_{k-1} * y_k < 0\}$.

Dans le cas où la liste comporte des éléments nuls, ce nombre, est par définition, égal au nombre de changements de signe entre les éléments de la même liste après suppression de tous les éléments nuls.

Exemple :

Le nombre de changements de signe entre les éléments de la liste $[1,0,-2,3,0,0,1,-1]$ est 3, car après suppression des zéros, on obtient la liste $[y_0, y_1, y_2, y_3, y_4] = [1,-2,3,1,-1]$ puisque $y_{k-1} * y_k < 0$ pour $k \in \{1,2,4\}$.

1. Ecrire une procédure Maple nommée ***Chgsgn*** qui prend en entrée une liste L dont les éléments sont de type numérique et retourne le nombre de changements de signe entre les éléments de L.

PARTIE II

Soit P un polynôme en x de degré n $(n > 2)$ à coefficients réels et à racines réelles simples.

On se propose de calculer le nombre de racines réelles de P sur un intervalle $[a,b]$.

On définit la suite de polynômes associée à P comme suit :

- P_0 est égal à P.
- P_1 est le polynôme dérivé de P ($P_1 = P'$).
- P_2 est l'opposé du reste de la division euclidienne de P_0 par P_1.
- P_3 est l'opposé du reste de la division euclidienne de P_1 par P_2.
- …
- P_n est l'opposé du reste de la division euclidienne de P_{n-2} par P_{n-1}.

2. Ecrire une procédure Maple nommée ***List_Poly*** qui prend en entrée un polynôme P et retourne la liste $[P_0, P_1, P_2, \ldots, P_n]$.

Soient a et b deux réels tels que $a < b$; Na et Nb représentent respectivement les nombres de changements de signe entre les éléments de la liste de polynômes $[P_0, P_1, P_2, \ldots, P_n]$ pour $x = a$ et $x = b$.

Le nombre de racines réelles de P sur un intervalle $[a,b]$ est égal à ($Na - Nb$), si P ne s'annule ni en a, ni en b.

Remarque : Pour $i \in [1,n]$, si P_i s'annule pour $x = a$ (ou P_i s'annule pour $x = b$), il n'intervient pas dans le décompte du nombre de changements de signe de la liste de polynômes $[P_0, P_1, P_2, \dots, P_n]$ pour $x = a$ et $x = b$.

3. Ecrire une procédure Maple nommée ***Nb_Chgs*** qui prend un polynôme $\boldsymbol{P}$ en x et un réel $\boldsymbol{r}$ et retourne le nombre de changements de signe entre les éléments de la liste $[P_0, P_1, P_2, \dots, P_n]$ pour $x = r$.

4. Ecrire une procédure Maple nommée ***Nb_Racine_Simple*** qui prend en entrée un polynôme $\boldsymbol{P}$ en x et deux réels $\boldsymbol{a}$ et $\boldsymbol{b}$ et retourne le nombre de racines réelles simples de P sur $[a,b]$.

Soit $\boldsymbol{P}$ un polynôme en x de degré $\boldsymbol{n}$ $(n > 2)$ à coefficients réels et à racines réelles simples et multiples.

5. On se propose de compter le nombre de racines simples et multiples du polynôme $\boldsymbol{P}$. Dans le cas où la racine est multiple elle sera comptabilisée une seule fois. Les racines de $\boldsymbol{P}$ sont les racines du polynôme $\dfrac{P}{\text{pgcd}(P, P')}$ n'ayant que des racines simples.

   Ecrire l'instruction Maple donnant le nombre de racines réelles de P sur $[a,b]$.

PROBLEME 2

Le but de ce problème est d'étudier un algorithme de chiffrement et de déchiffrement.

PARTIE I : CHIFFREMENT

Description

Soit un message de longueur n (avec n pair), écrit en lettres majuscules de A à Z, représenté par un tableau de taille n à raison d'une lettre par case.

L'idée du **chiffrement** est de grouper les lettres du message par bloc de 2, puis de les chiffrer bloc par bloc en utilisant la méthode suivante :

Les lettres sont codées par leur rang dans l'alphabet comme le montre le tableau suivant :

lettre	A	B	C	D	E	F	G	H	I	...	S	T	U	V	W	X	Y	Z
code	1	2	3	4	5	6	7	8	9	...	19	20	21	22	23	24	25	26

Tableau de codage

Les codes P_k et P_{k+1} de deux lettres successives d'un texte seront chiffrées C_k et C_{k+1} en utilisant la formule :

$$\begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} . \begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} \pmod{26}$$

avec :

- P_k et P_{k+1} respectivement les codes des lettres k et $k+1$ du texte clair (texte original non chiffré).
- C_k et C_{k+1} respectivement les codes des lettres k et $k+1$ du texte chiffré.
- La matrice $Mc = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ de chiffrement vérifiant les caractéristiques suivantes :
  - Les éléments a, b, c et d de Mc doivent être **des entiers positifs.**
  - La matrice Mc doit être inversible. La matrice inverse existe si $(ad-bc)$ est **impair et n'est pas multiple de 13**.

Exemple de chiffrement

On souhaite chiffrer le mot « **ELECTION** » représenté par le tableau suivant :

Mot à chiffrer	E	L	E	C	T	I	O	N

avec la matrice de chiffrement $Mc = \begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}$.

D'après le tableau de codage, les codes correspondants aux lettres du mot « **ELECTION** » sont :

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
5	12	5	3	20	9	15	14

Les deux premières lettres du mot seront donc chiffrées ainsi :

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 12 \end{pmatrix} \text{ (mod 26)} = \begin{pmatrix} 93 \\ 109 \end{pmatrix} \text{ (mod 26)} = \begin{pmatrix} 15 \\ 5 \end{pmatrix}$$

Les nombres 15 et 5 correspondent respectivement aux lettres O et E.
En procédant de même avec les paires de lettres suivantes, on obtient les codes chiffrés suivants :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
	15	5	5	20	8	7	9	17
Mot chiffré	O	E	E	T	H	G	I	Q

Travail demandé

On suppose avoir :

- effectué les déclarations suivantes :

  CONSTANTE NMAX = 1000
  TYPE TABMSG = tableau [1.. NMAX] de caractère
  TABCAR = tableau [1.. 26] de caractère
  TABE = tableau [1..12] de entier
  MATC = tableau [1..2 , 1..2] de entier

- défini le tableau constant T de type **TABCAR** suivant:

T[i]	A	B	C	D	E	F	G	H	I	...	S	T	U	V	W	X	Y	Z
i	1	2	3	4	5	6	7	8	9	...	19	20	21	22	23	24	25	26

1. Ecrire une fonction algorithmique ***Taille*** permettant de saisir et retourner un entier pair strictement positif et inférieur ou égal à un entier **NMAX** donné passé en paramètre.

2. Ecrire une fonction algorithmique ***Code_Car***, qui à partir d'un paramètre **c** de type caractère et du tableau constant **T**, retourne l'indice de la case contenant **c** de **T** si *c* appartient à **T** et 0 sinon.
3. Ecrire une procédure algorithmique ***Saisie_Msg*** permettant de saisir **n** lettres dans un tableau **T1** de type TABMSG en vérifiant l'appartenance au tableau **T** de chacune des lettres saisies.
4. Ecrire une procédure algorithmique ***Saisie_Matc*** permettant de saisir la matrice de chiffrement **Mc** de type MATC.
5. Ecrire une procédure algorithmique ***Chiffrer*** permettant de chiffrer dans un tableau **T2**, un message clair donné représenté par un tableau **T1** de taille *n*.

PARTIE II : DECHIFFREMENT

Pour déchiffrer un message, on multiplie les codes des lettres du message chiffré deux par deux par la matrice de déchiffrement.

$$\begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} . \begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} \pmod{26}$$

Avec :

- C_k et C_{k+1} respectivement les codes des lettres k et $k+1$ du texte chiffré.
- P_k et P_{k+1} respectivement les codes des lettres k et $k+1$ du texte clair.
- la matrice $Md = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1}$ est la matrice de déchiffrement. Le calcul de Md est donné par

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = m \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \bmod 26$$

où :

- $m \in \{1,3,5,7,9,11,15,17,19,21,23,25\}$
- m respecte la condition $((ad-bc)*m) \bmod 26 = 1$.

Exemple de déchiffrement

Considérant la matrice de chiffrement $\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}$ et calculant la matrice de déchiffrement correspondante : $Md = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}^{-1} = m \begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} \bmod 26$

La première valeur de m tel que $m \in \{1,3,5,7,9,11,15,17,19,21,23,25\}$ vérifiant la condition $((ad-bc)*m) \bmod 26 = 1$ est celle retenue. Dans ce cas cette valeur est égale à 23.

Ce qui donne : $Md = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = 23 \begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} \bmod 26 = \begin{pmatrix} 5 & 12 \\ 15 & 25 \end{pmatrix}$

On se propose de déchiffrer le mot « **IPHMKNDO** » représenté par le tableau suivant :

Mot à déchiffrer :								
	I	**P**	**H**	**M**	**K**	**N**	**D**	**O**

D'après le tableau de codage, les codes C_k correspondants au mot « **IPHMKNDO** » sont :

C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
9	16	8	13	11	14	4	15

Les deux premières lettres du mot chiffré seront donc déchiffrées ainsi :

$$\begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} 5 & 12 \\ 15 & 25 \end{pmatrix} \cdot \begin{pmatrix} 9 \\ 16 \end{pmatrix} \text{ (mod 26)} = \begin{pmatrix} 237 \\ 535 \end{pmatrix} \text{ (mod 26)} = \begin{pmatrix} 3 \\ 15 \end{pmatrix}$$

Les nombres 3 et 15 correspondent respectivement aux lettres C et O.

En procédant de même avec les paires de lettres suivantes, on obtiendra les codes P_k suivants :

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
	3	15	14	3	15	21	18	19
Mot déchiffré	C	O	N	C	O	U	R	S

<u>Travail demandé</u>

6. Ecrire une procédure algorithmique ***Init_Tab*** qui remplit, avec des entiers impairs de 1 à 26 et non multiples de 13, un tableau **E** de type TABE.
7. Ecrire une procédure algorithmique ***Cree_Md*** qui crée, à partir de la matrice **Mc** de chiffrement, la matrice **Md** de déchiffrement en utilisant le principe de calcul décrit précédemment.
8. Ecrire une procédure algorithmique ***Dechiffrer*** permettant de déchiffrer dans un tableau **T1**, un message chiffré donné représenté par un tableau **T2** de taille **n**.