



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

Concours Nationaux d'Entrée aux Cycles  
de Formation d'Ingénieurs  
Session 2013



الجمهورية التونسية  
وزارة التعليم العالي  
والبحوث العلمية  
المنظورات الوطنية للدخول  
إلى مراحل تكوين المهندسين  
دورة 2013

### Concours Toutes Options

Barème : EXERCICE : 3 points PROBLEME 1 : 6 points PROBLEME 2 : 11 points

### Alternative de correction de l'épreuve d'informatique

#### Barème sur 100

#### EXERCICE 1 (15 / 100)

1.5 pts par question

```
1. > rsolve({u(0)=-1,u(1)=-1,u(n+2)=(n+1)*u(n+1)-(n+2)*u(n)},u(n));  
2. > u:=unapply(%,n);  
3. > f:=unapply(sum(u(n)*x^n,n=0..infinity),x);  
4. > g:=unapply(sum(u(n)*x^n/n!,n=0..infinity),x);  
5. > K:=f@g;  
ou bien K:=unapply(f(g(x)),x);  
6. > evalf(K(-2),20);  
7. > limit(f(x),x=-infinity);limit(g(x),x=+infinity);  
8. > solve(diff(f(x),x,x)=0,x);  
ou bien solve(diff(f(x),x$2));  
9. > plot([f,g],-6..2,-2..10);  
ou bien plot([f(x),g(x)],x=-6..2,-2..10);  
10. > fsolve(f(x)=g(x),x);  
ou bien > solve(f(x)=g(x),x);  
evalf(%)
```

#### PROBLEME1 (30 / 100)

1. (5 pts)

```
> Cle_RSA:=proc(p::posint,q::posint)  
local z,n,c,d;  
z:=(p-1)*(q-1);  
n:=p*q;  
c:=nextprime(z);  
d:=c^(-1) mod z;  
return([n,c],[n,d]);  
end proc;
```

2. (3 pts)

```
> CryptDecryptE:=proc(a::posint,CL::  
list)  
local b;  
b:=a^op(2,CL) mod op(1,CL);  
return(b);  
end proc;
```

3. (5 pts)

```
> CryptDecryptList:=proc(L1::list  
,CL::list)  
local L2,i;  
L2:=[];  
for i from 1 to nops(L1) do  
L2:=[op(L2),CryptDecryptE(L1[i],  
CL)];od;  
return(L2);  
end proc;
```



4. (5 pts)

```
> Codage:=proc(CH::string)
local i,L;
L:=[];
for i from 1 to length(CH) do
L:=[op(L),Ord(CH[i])];od;
return(L);
end proc;
```

5. (5 pts)

```
> Decodage:=proc(L::list)
local i,CH;
CH:="";
for i from 1 to nops(L) do
CH:=cat(CH,Char(L[i]));od;
return(CH);
end proc;
```

6. (4 pts)

```
>CryptageC:=proc(CH::string,CL::list)
local L;
L:=codage(CH);
return(CryptDecryptList(L,CL));
end proc;
```

7. (3 pts)

```
>DecryptageC:=proc(L1::list,CL::list)
local CH,L;
L:=CryptDecryptList(L1,CL);
return(Decodage(L));
end proc;
```

## PROBLEME2 (55 / 100)

N.B: on accepte également la syntaxe E pour le passage par valeur et S ou E/S pour le passage par variable.

1. (2 pts)

```
Fonction Nombre
(min,max:entier):entier
variable e : entier
DEBUT
REPETER
Lire(e)
JUSQU'A e >= min ET e <= max
RETOURNER(e)
Fin
```

2. (3 pts)

```
Fonction Index(NBL : entier, Num : entier, Tnum :
TAB1) : entier
variable i,k : entier
DEBUT
i←1
k←-1
TANT QUE k = -1 ET i <= NBL FAIRE
SI Tnum[i] = Num ALORS k←i SINON i←i+1 FIN SI
FIN TANT QUE
RETOURNER(k)
FIN
```

3. (8 pts)

```
Procédure Ajout_Tit(VAR NBL :
entier, VAR Tnum : TAB1, VAR
Ttitre : TAB2, VAR Tnbexp : TAB1)
variable Nliv,Nbexpliv : entier
Tliv : chaîne
DEBUT
REPETER
Lire(Nliv)
JUSQU'A Index(NBL,Nl,Tnum) = -1
Lire(Tliv)
Nbexpliv←Nombre(1,NBEXP)
NBL←NBL + 1
Tnum[NBL]←Nliv
Ttitre[NBL]←Tliv
Tnbexp[NBL]←Nbexpliv
FIN
```

4. (2 pts)

```
Procédure Ajout_NbEX(NBL : entier,Tnum :
TAB1,Num,Nb : entier,VAR Tnbexp : TAB1)
variable k : entier
DEBUT
k←Index(NBL,Num,Tnum)
Tnbexp[k]←Tnbexp[k] + Nb
FIN
```



5. (3 pts)

Procédure Etat\_Bib(NBL : entier,  
Tnum : TAB1, Ttitre : TAB2,  
Tnbexp : TAB1)  
variable i : entier  
Début  
POUR i de 1 à NBL FAIRE  
SI Tnbexp[i] > 0 ALORS  
Ecrire(Tnum[i], Ttitre[i], Tnbexp[i])  
FIN SI  
FIN POUR  
FIN

6. (3 pts)

Fonction Aut\_Emp(NCIN : entier, NBEMP : entier, M :  
MAT, NP : entier, P : TAB3) : booléen  
variable i : entier  
B1, B2 : booléen  
DEBUT  
i ← 1  
B1 ← faux  
TANT QUE B1 = faux ET i ≤ NBEMP FAIRE  
SI NCIN = M[i, 1] ALORS B1 ← vrai SINON i ← i + 1  
FINSI  
FIN TANT QUE  
i ← 1  
B2 ← faux  
TANT QUE B2 = faux ET i ≤ NP FAIRE  
SI NCIN = P[i] ALORS B2 ← vrai SINON i ← i + 1 FINSI  
FIN TANT QUE  
RETOURNER(B1 ET B2)  
FIN

7. (4 pts)

Procédure Calcul\_Date(JE, ME :  
entier, VAR JR, MR : entier)  
DEBUT  
JR ← JE + 10  
MR ← ME  
SI JR > 31 ET (ME = 1 OU ME = 3  
OU ME = 5 OU ME = 7 OU ME = 8  
OU ME = 10 OU ME = 12) ALORS  
JR ← JR - 31  
MR ← MR + 1  
FIN SI  
SI JR > 30 ET (ME = 4 OU ME = 6  
OU ME = 9 OU ME = 11) ALORS  
JR ← JR - 30  
MR ← MR + 1  
FIN SI  
SI JR > 28 ET (ME = 2) ALORS  
JR ← JR - 28  
MR ← 3  
FIN SI  
FIN

8. (12 pts)

Procédure Emprunt(NP : entier, P : TAB3, VAR  
NBEMP : entier, VAR M : MAT, VAR Tnbexp : TAB1)  
variable NCIN, Num, i, k, JE, ME : entier  
B : booléen  
DEBUT  
NCIN ← Nombre(10000000, 99999999)  
B ← faux  
i ← 1  
TANT QUE B = faux ET i ≤ P FAIRE  
SI P[i] = NCIN ALORS B ← vrai SINON i ← i + 1 FIN SI  
FIN TANT QUE  
SI B = faux ALORS  
REPETER  
Lire(Num)  
k ← Index(NBL, Num, Tnum)  
JUSQU'A k < 0  
SI Tnbexp[k] > 0 ALORS  
Lire(JE)  
Lire(ME)  
Tnbexp[k] ← Tnbexp[k] - 1  
NBEMP ← NBEMP + 1  
M[NBEMP, 1] ← NCIN  
M[NBEMP, 2] ← Num  
M[NBEMP, 3] ← JE  
M[NBEMP, 4] ← ME  
Calcul\_Date(JE, ME, M[NBEMP, 5], M[NBEMP, 6])  
M[NBEMP, 7] ← faux  
SINON Ecrire("livre non disponible") FIN SI  
SINON Ecrire("non autorisé à emprunter") FIN SI  
FIN





9. (3 pts)

Procédure Maj\_Pen(NBEMP, Jo, Mo : entier, VAR M : MAT)  
variable i : entier  
DEBUT  
POUR i de 1 à NBEMP FAIRE  
SI (M[i,6] > Mo) OU (M[i,6] = Mo  
ET M[i,5] < Jo) ALORS M[i,7] ← 1  
FIN SI  
FIN POUR  
FIN

10. (6 pts)

Procédure Etat\_Pen(NBEMP, Jo, Mo : entier, M : MAT)  
variable i : entier  
DEBUT  
Maj\_Pen(NBEMP, Jo, Mo, M)  
POUR i de 1 à NBEMP FAIRE  
SI (M[i,6] > Mo) OU (M[i,6] = Mo ET M[i,5] < Jo)  
ALORS  
k ← Index(NBT, M[i,2], Tnum)  
Ecrire( M[i,1])  
Ecrire(Ttitre[k])  
FIN SI  
FIN POUR  
FIN

11. (9 pts)

Procédure Retour(NBT, Num, Jo, Mo : entier, Tnum :  
TAB1, VAR NP, NBEMP : entier, VAR M : MAT,  
VAR Tnbexp : TAB1, VAR P : TAB3)  
Variable  
Début  
Maj\_Pen(NBEMP, Jo, Mo, M)  
i ← 1  
Tant que M[i,2] <> Num ET i <= NBEMP Faire  
i ← i + 1  
Fin tant que  
Si M[i,7] = 1 alors  
NP ← NP + 1  
P[NP] ← M[i,1]  
Fin si  
k ← Index(NBT, Num, Tnum)  
Tnbexp[k] ← Tnbexp[k] - 1  
Si i = NBEMP alors NBEMP ← NBEMP - 1  
Sinon  
Pour j de 1 à 7 faire  
M[i,j] ← M[NBEMP, j]  
Fin pour  
Fin si  
Fin