



Concours Nationaux d'Entrée aux Cycles de Formation d'Ingénieurs
Session 2011

Concours Toutes Options
Corrigé de l'épreuve d'Informatique

Barème : EXERCICE 1 : 4.5 points
EXERCICE 2 : 5.5 points
EXERCICE 3 : 4.5 points
EXERCICE 4 : 5.5 points

Le barème est sur 40

EXERCICE 1 (9 pts = 1 pt par question)

1. $> f:=(a,x) \rightarrow (x^2 - a \cdot x) \cdot \exp(1/x)$; ou bien
 $> f:=unapply((x^2 - a \cdot x) \cdot \exp(1/x), (a, x));$
2. $> series(f(a, x), x=infinity, 3);$
3. $> limit(f(a, x), x=0, left); limit(f(a, x), x=0, right);$
4. 1ère manière: $> df:=D[2](f);$
2ème manière : $> df:=unapply(diff(f(a, x), x), (a, x));$
5. $> solve(df(a, x)=0, x)$; ou bien
 $> solve(diff(f(a, x), x)=0, x);$ # (= 0 est facultative)
6. $> plot([seq(f(k, x), k=1..5)], x=-5..5, -10..10)$; ou bien
 $> plot([f(1, x), f(2, x), f(3, x), f(4, x), f(5, x)], x=-5..5, -10..10);$
7. $> a1:=solve(df(a, x), a);$
8. $> plot(f(2, x), x=-10..10, -10..10);$
9. $> simplify(diff(f(2, x), x));$

EXERCICE 2 (11 pts)

A. (2.5 pts = 0.25 pts par question)

- A.1. $> P:=(X-1) \cdot (X^2+1)^5;$
- A.2 $> degree(P, X)$; ou bien $degree(P)$;
- A.3. $> sort(expand(P));$



A.4. > seq(coeff(P,X,i),i=1degree(P)..degree(P));

A.5. > quo(P,a*X^3+b,X);rem(P,a*X^3+b,X);

B. (3 pts)

> Td:=array(1..50):

 j:=1;

 for n from 3 to 50 do

 S:=sum(X**k,k=0..n);

 if divide(S,1+X+X**2) then Td[jj]:=degree(S);j:=j+1 fi;od;

C. (1.5 pts)

C.1. > R:=n-> product(X^k+1,k=1..n); 1 pt

C.2 > lcoeff(R(50)); 0.5 pts

D. (4 pts)

> MULTIPLICITE:=proc(P::polynom,a::numeric)

 local B,k;

 if subs(X=a,P)<>0 then return(0)

 else

 k:=0;

 do

 k:=k+1;

 B:=evalb(simplify(subs(X=a,diff(P,X\$k)))=0);

 if B=false then break fi; od;

 return(k+1);fi;

 end proc;

EXERCICE 3 (9 pts)

1 point

1. Procédure AFF_QUES (k : entier , Ques : TABQ)

 Variable i : entier

 Début

 Pour i de 1 à k Faire

 Ecrire (i , Ques [i])

 Fin Pour

 Fin

3 points

2. Procédure REPONSES (k : entier , Ques : TABQ , VAR Rep : TABR)

Variable num , R : entier

V : caractère

Début

Répéter

Répéter

Ecrire ("donner le numéro de la question : ")

Lire (num)

Jusqu'à num >= 1 ET num <= k ET REP[num] = -1

Ecrire (Ques [num])

Répéter

Ecrire ("donner votre réponse : ")

Lire (R)

Jusqu'à R = 1 OU R = 0

Répéter

Ecrire ("voulez-vous valider ")

Lire (V)

Jusqu'à V = "O" OU V = "N"

Si V = "O" Alors REP[num] ← R Fin Si

Répéter

Ecrire ("voulez vous répondre à une autre question : ")

Lire (V)

Jusqu'à V = "O" OU V = "N"

Jusqu'à V = "N"

Fin

2 points

3. Fonction TOTAL (k : entier , Rep : TABR , Correct : TABC) : entier

Variable i , S : entier

Début

S ← 0

Pour i de 1 à k Faire

```

    Si (Rep [i] = 1 ET Correct [i]=FAUX) OU (Rep [i] = 0 ET Correct[i]) Alors
        S ← S - 1
    Sinon
        Si (Rep [i] = 1 ET Correct [i]) OU (Rep [i] = 0 ET Correct [i] = FAUX)
            Alors S ← S + 2
        Fin Si
    Fin Pour
    Retourner (S) Fin

```

3 points

4. Algorithme INTERROGATION

```

TYPE TABQ = tableau [1..100] de chaîne de caractère
    TABR = tableau [1..100] de entier
    TABC = tableau [1..100] de booléen
Variable Ques : TABQ
    Rep : TABR
    Correct : TABC
    N , NMAX , i : entier
Début
    REPETER
        Ecrire ("donner le nombre de questions : ")
        Lire (N)
        JUSQU'A N >= 2 ET N <= NMAX
        Pour i de 1 à N Faire
            REP[i] ← -1
        Fin Pour
        AFF_QUES(N , QUES)
        REPONSES(N , QUES , REP)
        T ← TOTAL(N , REP , CORRECT)
        Si T <= 0 Alors Ecrire (" la note obtenue sur 20 est : 0 ")
            Sinon Ecrire (" la note obtenue sur 20 est : " , (T*20) / (2*N))
        Fin Si
    Fin

```

EXERCICE 4 (11 pts)

1 point

1. Procédure Saisie_R (R : MATR)

```

Variable i , j : entier
Début
    Pour i de 1 à N Faire

```

Pour j de 1 à K Faire

Si i=j Alors R[i,j] \leftarrow -1

Sinon Répéter Lire(R[i,j]) Jusqu'à R[i,j] = 0 OU R[i,j] = 1 OU R[i,j] = 2

Fin Pour

Fin Pour

Fin

points

Fonction Nb_PTS (R : MATR, k : entier, AR :booleen) : entier

Variable i, Np : entier

Début

Np \leftarrow 0

Pour i de 1 à N Faire

Si R[k, i] = 1 Alors Np \leftarrow Np + 1

Sinon Si R[i, k] = 2 Alors Np \leftarrow Np + 3 Fin Si Fin Si

Si R[i, k] = 1 Alors Np \leftarrow Np + 1

Sinon Si R[i, k] = 2 Alors Np \leftarrow Np + 3 Fin Si Fin Si

Fin Pour

Retourner (S)

Fin

point

Procédure TABL_PTS (R : MATR, VAR T : TABP)

Variable k : entier

Début

Pour k de 1 à N Faire

T[k] \leftarrow Nb_PTS(R, k)

Fin Pour

Fin

oint

Procédure Saisie_B (B : MATB)

Variable i, j : entier

Début

Pour i de 1 à N Faire

Pour j de 1 à 2 Faire

Lire(B[i , j])

Fin Pour

Fin Pour

Fin

Procédure CHAMPIONS(R : MATR, T : TABP, B : MATB, VAR d : entier, VAR CH : MATB)

Variable

Début

MAXP \leftarrow T[1]

Pour i de 2 à N Faire

Si T[i] > MAXP Alors MAXP \leftarrow T[i] Fin Si

Fin Pour

d \leftarrow C

Pour i de 1 à N Faire

Si T[i] = MAXP Alors

d \leftarrow d + 1

CH[d , 1] \leftarrow i

CH[d , 2] \leftarrow B[i , 1] - B[i , 2]

Fin Si

Fin Pour

Fin

PTS

Procédure AFF_CHAMPIONS (R : MATR, T : TABP, B : MATB)

Variable i , j , d : entier

C : MATB

Début

CHAMPIONS(R , T , B , d , C)

Pour i de 1 à d Faire

j \leftarrow C[i , 1]

Ecrire (" Numéro Equipe total points total points aller total points
retour écarts entre buts marqués et buts encaissés nombre de buts
marqués ")

Ecrire (j , T[j] , Nb_PTS (R , j , VRAI) , Nb_PTS (R , j , FAUX) , C[j , 2] , B[j , 1])

Fin Pour Fin