

Proposition de corrigé

Concours : Concours Commun Polytechniques

Année : 2016

Filière : TSI

Épreuve : Informatique

Ceci est une proposition de corrigé des concours de CPGE, réalisée bénévolement par des enseignants de Sciences Industrielles de l'Ingénieur et d'Informatique, membres de l'[UPSTI](http://www.upsti.fr) (Union des Professeurs de Sciences et Techniques Industrielles), et publiée sur le site de l'association :

<https://www.upsti.fr/espace-etudiants/annales-de-concours>

A l'attention des étudiants

Ce document vous apportera des éléments de corrections pour le sujet traité, mais n'est ni un corrigé officiel du concours, ni un corrigé détaillé ou exhaustif de l'épreuve en question.

L'UPSTI ne répondra pas directement aux questions que peuvent soulever ces corrigés : nous vous invitons à vous rapprocher de vos enseignants si vous souhaitez des compléments d'information, et à vous adresser à eux pour nous faire remonter vos éventuelles remarques.

Licence et Copyright

Toute représentation ou reproduction (même partielle) de ce document faite sans l'accord de l'UPSTI est **interdite**. Seuls le téléchargement et la copie privée à usage personnel sont autorisés (protection au titre des [droits d'auteur](#)).

En cas de doute, n'hésitez pas à nous contacter à : corrigesconcours@upsti.fr.

Informez-vous !

Retrouvez plus d'information sur les [Sciences de l'Ingénieur](#), l'[orientation](#), les [Grandes Ecoles](#) ainsi que sur les [Olympiades de Sciences de l'Ingénieur](#) et sur les [Sciences de l'Ingénieur au Féminin](#) sur notre site : www.upsti.fr

L'équipe UPSTI

GESTION DE DÉCHETTERIES

Correction UPSTI

Remarque : les réponses en Scilab aux différentes questions de programmation sont regroupées à la fin du corrigé.

PARTIE 1 - Étude du système

Question 1

Pas de réelles difficultés, une question originale, qui invite à traduire de manière simple (mais incomplète) un traitement séquentiel.

Réponse

```
Fonction EntréeUtilisateur()
```

```
Début
```

```
  Tant que (badge non présent) faire  
    afficherMessage("passez votre badge dans le lecteur")  
  Fin Tant que.
```

```
  codeRfid ← lireBadge( ) ;
```

```
  si ( (authentifier(codeRfid) ET ( nbUtilisateur < NbMAX) ET  
      (nonDétecterPrésencePontPesage() ))
```

```
  alors
```

```
    début
```

```
      CommanderBarriere(3, 1) ; #ouvrir barrière3  
      IncrementerUsager( ) ;
```

```
      Tant que (nonDétecterPrésencePontPesage() ) faire
```

```
        Attendre( ) ;
```

```
      finTant que
```

```
      CommanderBarriere(3, 0) # fermer barrière 3
```

```
      Masse ← peser( ) ;  
      mémoriser(codeRfid , masse) ;
```

```
      #descente du pont en entrant  
      CommanderFeu(5,0) ;
```

```
      Tant que ( DétecterPrésencePontPesage() ) faire
```

```
        Attendre( ) ;
```

```
      fin Tant que
```

```
      CommanderFeu(5,2) ;
```

```
    fin
```

```
  sinon
```

```
    AfficherMessage("veuillez attendre ou vous authentifier")
```

```
  finsi
```

```
fin
```

PARTIE 2 - Étude de l'authentification de l'utilisateur de la déchetterie

Question 2

Codage décodage Hexa-> binaire ->Hexa

Fonction logique (OUX)

Réponse		
identifiant du lecteur \$50		
décodage	Hexa -> binaire	
calcul de checksum	50 ← 0101 0000	
	03 → 0000 0011	
	25 → 0010 0101	
	26 → 0010 0110	
	01 → 0000 0001	
	opération ou exclusif résultat	0101 0001 → 51
la trame s'écrit donc bien <AA 50 03 25 26 01 51 BB>		

Question 3

On demande ici, de commenter un algorithme, On n'attend pas uniquement une description de chaque commande mais de d'expliquer la fonction des deux boucles.

Réponse Algorithme à commenter

```
/* En pseudo langage, on considèrera que les indices commencent à 1.*/
var :
trameLectureBadge[AA,50,03,25,26,01,51,BB] ;
trameRéponseBadge[10] ;
checksum=00 ; /*Hexadécimal*/
num_badge=0;

Début programme
  canal → configurer_Communication_avec_lecteur_RFID() ;
  ouverture du canal de transmission(canal) ;

  /*envoi trame requête*/
  écrireSurPortSérie (trameLectureBadge , portSérie ) ;

  /*réception trame réponse*/
  trameRéponseBadge ← lectureSurPortSérie(10) ;

  /*calcul du checksum*/
  Nb_octet ← trameRéponseBadge[3] ;
  /*lire nb octet numéro de carte+ statut */

  Pour i de 2 à Nb_octet+2 inclus faire
    /* variation indice i pour calcul checksum*/
    checksum ← checksum OUEX trameRéponseBadge[i] ;
    /* calcul du checksum par ou exclusif */
  fin pour
  /* explication itération Pour : on calcule le checksum sur l'ensemble de la
  trame retour en enlevant le premier octet (0xAA) et le dernier (0xBB)
  Initialisation à 0 du checksum, puis ouexclusif octet par octet */

  /* Vérification du checksum */
  si (checksum= trameRéponseBadge[3+Nb_octet] )
    /* comparaison checksum calculé avec celui de la trame */
    alors
      /* calcul du numéro de badge */
      Pour i de 5 à nb_octet+2 inclus faire
        /* indice de 5 à nb_octet +2 pour récupérer les 4 octets du numéro*/
        j ← nb_octet+2-i ; /* calcul indice du poids de l'octet */
        num_badge ← num_badge +( trameRéponseBadge[i]*16puissance(j) ) ;
        /*numéro de badge : addition des 4 valeurs hexa avec les 4
        poids pour obtenir un nombre en décimal */
      fin pour
      /* explication itération Pour : pour l'exemple donné i (indice de liste)
      varie de 5 à 8 alors que j (indice représentant le poids de l'octet) varie
      de 3 à 0, puis addition pour obtenir un nombre décimal */

      sinon afficher ("votre badge n'est pas reconnu " ) ;
    fin si
  fin si

Fin programme
```

Question 4 en Python

La traduction en langage (python ou scilab) ne pose pas de pb

La demande doit être écrite en hexa

Compléter la commande d'ouverture série nécessite d'aller rechercher dans la documentation, l'explication des différents paramètres.

Réponse

```
import serial
#les variables
Demande= [0xAA,0x50,0x03,0x25,0x26,0x01,0x51,0xBB]

CS=0
Num_Badge =0
#configuration de la liaison
Ser=serial.Serial(COM1,19200,8,parity='N',2)
Ser.open()

#envoi de la requête
Ser.write(Demande)

#réception de la réponse
reponse= Ser.read(10)

#calcul du checksum
nb_octet=reponse[2]
checksum=0x00
for i in range(1,nb_octet+2):
    checksum=checksum^reponse[i]

#calcul du numéro de badge reçu si le checksum correspond
if checksum==reponse[nb_octet+2]:

    for i in range(4,nb_octet+2):
        j=nb_octet-i+1
        num_badge=num_badge+reponse[i]*(16**j)

else:
    print("erreur de communication")
```

Question 5

Rien de particulier, nécessite de connaître les structures des bases de données et de bien lire le sujet

Réponse

Applicatif déployé sur ordinateur industriel : **client**

Applicatif déployé sur ordinateur bureautique : **serveur**

Question 6

La lecture de l'algorithme n'est pas aisée, en s'appuyant sur le tableau à compléter on comprend le fonctionnement de l'algo.

Réponse
Voir tableau ci-dessous.
Rôle de l'algorithme : Réaliser un tri croissant de la liste

i	Nb de boucles Tant que	Liste à la fin de la boucle Pour				
État initial		12020501	12020508	12020506	12020502	12020504
2	1	12020501	12020506	12020508	12020502	12020504
3		12020501	12020502	12020506	12020508	12020504
4		12020501	12020502	12020504	12020506	12020508
5						

Une simulation du fonctionnement est disponible sur le site :

<http://goo.gl/h2o2Uy>

Question 7

On demande ici de rechercher le variant de la boucle (plus facile que l'invariant -voir plus bas le commentaire-). Cela nécessite de comprendre le fonctionnement de la boucle

Réponse
Variable : i
Justification :
La variable i est incrémentée à chaque évolution de la boucle. (son évolution est monotone).
N est supérieur à i à l'entrée de la boucle et l'évolution de i étant monotone croissante, la condition $i \leq n$ implique qu'à terme la condition d'arrêt $i > n$ sera vérifiée:

Note supplémentaire sur l'invariant de boucle

L'assertion $\star\star \text{tab}[1..i-1]$ est trié $\star\star$ est l'invariant de boucle

Elle est vérifiée initialement, pour $i = 2$, et pourvu que l'insertion se fasse correctement, on peut affirmer que sous cette précondition, après chaque étape, la zone $\text{tab}[1..i]$ sera triée, donc que l'invariant sera vérifié pour $i=i+1$, c.-à-d. à nouveau en début de l'étape suivante.

L'algorithme ne termine que si la condition d'arrêt $i > n$ est vérifiée ; on a alors $i=n+1$ (l'incréméntation de i n'ayant lieu que si $i=n$), et donc que $\text{tab}[1..n]$ est trié : l'algorithme est partiellement correct.

Question 8

Un calcul classique de complexité, avec deux boucles imbriquées, le mot linéarithmique dans le sujet peut avoir troublé les candidats
(https://fr.wikipedia.org/wiki/Analyse_de_la_complexit%C3%A9_des_algorithmes)

Réponse

Classe de complexité : **quadratique, $\Theta(n^2)$.**

Dans le meilleur des cas, avec des données déjà triées, l'algorithme effectuera seulement n comparaisons. Sa complexité dans le meilleur des cas est donc en $\Theta(n)$.

Dans le pire des cas, avec des données triées à l'envers, les parcours successifs du tableau imposent d'effectuer $(n-1)+(n-2)+(n-3)..+1$ comparaisons et échanges, soit $(n^2-n)/2$. On a donc une complexité dans le pire des cas du tri par insertion en $\Theta(n^2)$.

Si tous les éléments de la série à trier sont distincts et que toutes leurs permutations sont équiprobables, la complexité en moyenne de l'algorithme est de l'ordre de $(n^2-n)/4$ comparaisons et échanges. La complexité en moyenne du tri par insertion est donc également en $\Theta(n^2)$.

Question 9

Algorithme de recherche récursif

Réponse

Particularité de l'algorithme : **C'est un algorithme récursif !**

Précaution(s) sur m et n : **$m \geq 1$; $n \leq \text{longueur}(\text{tab})$; $m \leq n$.**

Question 10

Pour l'algorithme, on peut proposer d'autres solutions dont une recherche classique d'une valeur dans un tableau.

Réponse

recherche(x,tab)

Fonction permettant de rechercher l'occurrence de x dans tab[]

Entrées : x : entier, occurrence à rechercher.
tab[] : tableau d'entiers

Sortie : entier : position de l'occurrence du nombre recherché si trouvé, None sinon

Début

```
g ← 1
d ← longueur(tab)
Tant que g ≤ d alors,
  m ← Division entière de (g+d) par 2
  Si (tab[m]=x)
    alors
      Retourner m
  Sinon
    Si tab(m)<x
      alors
        g ← m+1
    Sinon
      alors
        d ← m+1
    Fin si
  Fin si
Fin Tant que
Retourner None
Fin fonction
```

Réponse

Place dans la pile :

La solution itérative explicite n'utilise pas de pile, elle est donc moins gourmande en espace.

PARTIE 3 - Étude du bloc logiciel de traitement du signal intégré au peson.

Question 11

Lecture pas très facile, beaucoup de bruit

Réponse

Valeur de la fréquence minimale d'échantillonnage :

Fmini=2*fmax=100Hz

Question 12

C'est assez facile d'identifier l'allure générale du premier ordre (la question posée) par contre pour les paramètres des fonctions sinusoïdales, il faut faire confiance au sujet !

Réponse

Justifier les valeurs des paramètres retenus :

La réponse temporelle a globalement l'allure d'un premier ordre (pas de dépassement hors le bruit, et la tangente à l'origine semble non nulle)

- Le temps de réponse est alors de $3 \cdot \tau = 1500s$,
- La valeur finale est d'environ 5300,

Question 13

RAS

Réponse

$$x_1(t) = x_f(1 - e^{(-t/\tau)}) + a_1 \sin(2.\pi.f_1.t + \varphi_1) + a_2 \sin(2.\pi.f_2.t + \varphi_2) + a_3 \sin(2.\pi.f_3.t + \varphi_3)$$

Question 14 en Python

ras

Réponse

```
# définition du signal de test
def signal(t):
    return ( 200.0*math.sin(10*math.pi*t) \
            +200.0*math.sin(20*2*math.pi*t+math.pi/3) \
            +200.0*math.sin(50*2*math.pi*t+math.pi/5) \
            +5320.0*(1-math.exp(-t/0.5)) )
```

Question 15 en Python

RAS

Réponse

```
def bruitGauss(sigma):
    U1 = random.random()
    U2 = random.random()
    return ( sigma*math.sqrt(-2*math.log(U1))*math.cos(2*math.pi*U2) )
```

Question 16 en Python

Utilisation des deux fonctions décrites au dessus

Réponse

```
# figure du signal de test
from matplotlib.pyplot import *

fe=500
T=5.0
N=int(fe*T) (ou 2500)
te = T/N
x2 = numpy.zeros(N)
t = numpy.zeros(N)
ampbruit=500
sigma = 0.3
for k in range(N):
    t[k] = k*te
    x2[k] = signal(t[k])+ampbruit*bruitGauss(sigma)

figure(figsize=(12,6))
plot(t,x2)
xlabel('temps [s]')
ylabel('signal de test x2 [kg]')
grid()
title('le signal de test')
```

Question 17 en Python

Bien comprendre les deux boucles imbriquées, pour générer les différents termes

Réponse

```
# fonction convolution

def convolution(H,x):
    N=H.size
    nx=x.size
    ny= nx-(N-1)
    y=numpy.zeros(ny)
    for n in range(ny):
        for k in range(N):
            y[n]+=H[k]*x[(N-1)+n-k]
    return y
```

Question 18 en Python

Réponse

```
# filtre gaussien
P =200
h = numpy.zeros(2*P +1)
epsilon=0.0001
sigma=P/math.sqrt(-2.0*math.log(epsilon))
som = 0.0
for k in range(2*P+1):
    h [k] = math.exp(-(k-P)**2/(2*sigma**2))
    som += h [k]
h = h /som
```

Question 19

Réponse

Le signal obtenu est donc lissé et est représentatif de la masse
Par contre on observe un temps de retard, temps à prendre en compte dans la suite du programme

PARTIE 4 - Étude de la base de données du SIVOM

Question 20

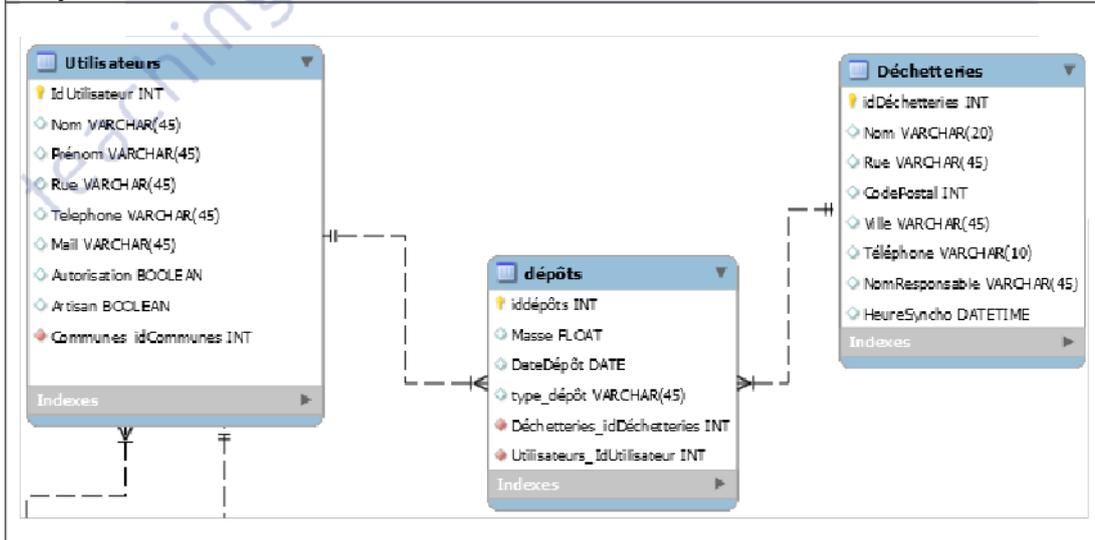
Bien lire le sujet

Réponse

Les données enregistrées dans le PC du poste sont transmis à la gestion centralisée **chaque nuit** : les id_utilisateurs, la masse déposée, le type de dépôt

Question 21

Réponse



Identifiant(s)	idDepots	idDechetterie
Clé(s) étrangère(s)	IdUtilisateur	IdDechetterie

Question 22

Une requête simple pour débiter

Réponse		
Requête SQL :		
<pre>Select nom, mail From utilisateurs where artisan ='oui' and autorisation ='oui'</pre>		
le résultat	nom PIERRE	mail <u>daniel.pierre@mail.fr</u>

Question 23

Ajout d'un élément dans une table,

Réponse		
Condition : la commune du nouveau déposant doit exister dans la table Communes avant l'ajout dans la table utilisateurs		
requête SQL :		
<pre>INSERT INTO Utilisateurs(Nom, Prénom, Rue, Telephone, Mail, Autorisation, Artisan, idCommunes) VALUES ('Dupont', 'Christian', 'Rue de la poste', 'Telephone', 'Mail', 'oui', 'Non', 15)</pre>		

Question 24

Réponse			
Requête SQL :			
<pre>SELECT nom, prenom, mail FROM Utilisateurs, Communes WHERE Utilisateurs.autorisation='oui' AND utilisateurs.communeIDCommune = Communes.IDCommune AND Communes.nom='SENS' ORDER BY nom, prenom</pre>			
Résultat :	nom	prénom	mail
	Dupond,	jean,	Jean.dupond@mail.fr
	Daniel.pierre@mail.fr		PIERRE Daniel

Question 25

Réponse

Requête SQL

```
SELECT ID_dechetterie as DECHETTERIE, count(*) as nbDepotVert,  
sum(masse) as masseTotaleVert, sum(masse)*10 as recette  
FROM `depots`  
WHERE `type depot`='vert'  
and `date depot` > TIMESTAMP('2015-01-01')  
and `date depot` < TIMESTAMP('2015-12-31')  
GROUP BY `ID_dechetterie`
```

for innovation



teaching sciences

Question 4 en Scilab

Réponse

```
//les variables
demande=[170,80,3,37,38,1,81,187]
checksum=0;
num_badge=0;

// configuration de la liaison
Ser=openserial (COM1,19200,parity='N',8,2)

//envoi de la requête
writeseial( Ser , Demande)

sleep(1000);
// réception de la réponse
réponse= readserial( Ser,10)

//calcul du checksum
nb_octet=reponse(3);
printf("\n nb ocete =%d',nb_octet);

for i=2:(nb_octet+2)
    checksum=bitxor( checksum, reponse(i) );
end
printf ("\n checksum= %d',checksum);

//calcul du numéro de badge reçu si le checksum correspond
if (checksum==reponse(nb_octet+3)) then

    for i=5:nb_octet+2
        j=nb_octet-i+2;
        num_badge=num_badge+reponse(i)*(16**j) ;
    end
    printf("\n valeur du badge : %d',num_badge);
else
    printf("\n erreur de communication")
end
```

Question 14 en Scilab

Réponse

```
function x1=signal(t)
    x1=200.0*sin(10*%pi*t)..
        +200.0*sin(20*2*%pi*t+%pi/3)..
        +200.0*sin(50*2*%pi*t+%pi/5)..
        +5320.0*(1-exp(-t/0.5));
endfunction
```

Question 15 en Scilab

Réponse

```
function bruit=bruitGauss(sigma)
    rand("uniform");
    U1 = rand();
    U2 = rand();
    bruit = sigma*sqrt(-2*log(U1))*cos(2*pi*U2);
endfunction
```

Question 16 en Scilab

Réponse

```
fe = 500;
T = 5.0;
N = int(fe*T);
te = T/N;
x2 = zeros(N,1);
t = zeros(N,1);
ampbruit = 500;
sigma = 0.3;
for k=1:N
    t(k) = k*te;
    x2(k) = signal(t(k))+ampbruit*bruitGauss(sigma);
end
figure(0);
plot(t,x2);
xlabel("Temps [s]");
ylabel("Masse [kg]");
set(gca(),"grid",[1 1]);
title("Le signal de test");
```

Question 17 en Scilab

Réponse

```
function y=convolution(H,x)
    N = size(H, "r");
    nx = size(x, "r");
    ny = nx-(N-1);
    y = zeros(ny,1);
    for n=1:ny
        for k=1:N
            y(n) = y(n)+H(k)*x(N+n-k);
        end
    end
endfunction
```

Question 18 en Scilab

Réponse

```
// Q18 filtre gaussien
P = 200;
h = zeros(2*P+1,1);
nh = size(h, "r");
epsilon = 0.0001;
sigma = P/sqrt(-2.0*log(epsilon));
som = 0.0;
for k=1:2*P+1
    h (k) = exp(-(k-P)**2/(2*sigma**2));
    som = som + h (k);
end
h = h /som;

// suite du programme pas demandée

figure(3);
plot(h);
xlabel("Indices n");
ylabel("Valeurs des coefficients de H");
xgrid();
title('Filtre Gaussien');

// vérification de l'effet sur le signal
y2 = convolution(h,x2); // Filtre gaussien
ny2 = size(y2, "r");
ty2 = zeros(ny2,1);
for k=1:ny2
    ty2(k) = (nh-1)*te+te*k;
end
figure(4);
plot(t,x2);
set(gca(),"auto_clear","off");
plot(ty1,y1,"g");
plot(ty2,y2,"r");
xlabel("Temps [s]");
ylabel("Masse [kg]");
xgrid();
title('Effet du filtre gaussien');
```