IQUETTE
NTIFICATION

DOUANES
& DROITS
INDIRECTS

Concours pour l'emploi de :

Contrôleur Programmeur

Date : 23 / 02 / 22

Epreuve écrite n° 2

Matière : Algorithme et écriture de programme

Nombre d'intercalaire(s) joint(s) → 8

A L'ATTENTION DU CANDIDAT

lit aussi bien de signer à la fin de la composition que d'indiquer u tout signe distinctif sur les feuilles intercalaires.

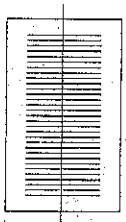
nettre à l'administration d'identifier votre copie, collez sur cette page "Code à barres" aux emplacements prévus à cet effet.

POSITIONNEMENT DES ÉTIQUETTES

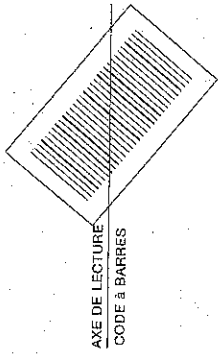
ttre la lecture optique de l'étiquette, le trait vertical matérialisant l'axe u code à barres doit traverser la totalité des barres de ce code.

EXEMPLES

BON



MAUVAIS



A

NOTE / 20

18,00

Paraphe correcteur

question 1 /

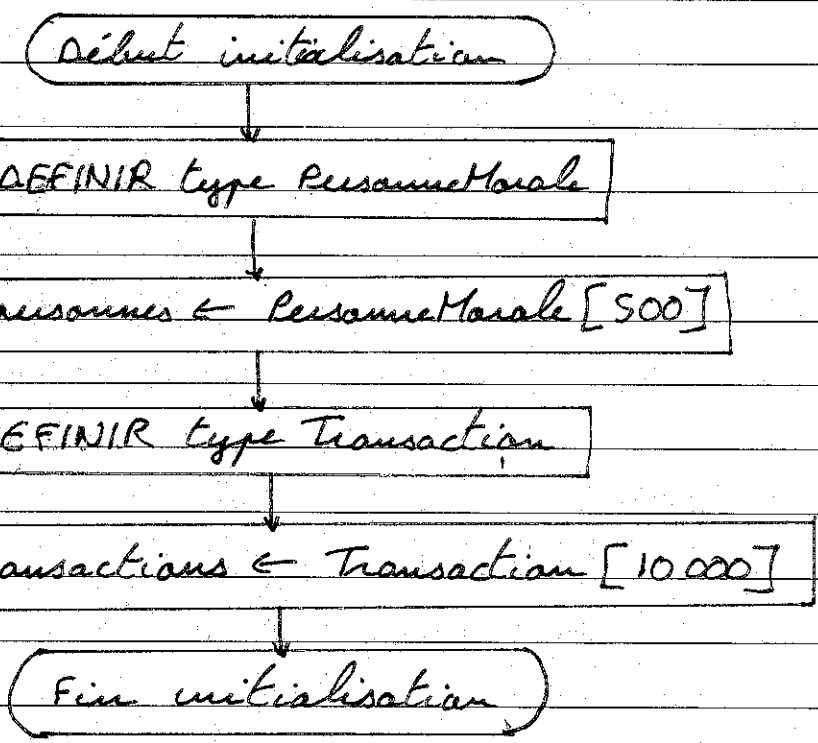
On peut placer ces identifiants dans des variables plus explicites.

question 2 /

// J'utilise gold pour eviter la confusion avec OR.

```
gold = "A"
platine = "P"
argent = "a"
palladium = "p"
```

question 3 /



Question 4 /

```
class PersonneMorale :

    def __init__(nom = None, adresse=None):
        self.nom = nom
        self.adresse = adresse

personnes = []
for _ in range(500):
    personnes.append(PersonneMorale())

class Transaction :

    def __init__(metal = None,
                 quantite = 0,
                 prix = 0,
                 identifiant_vendeur = None,
                 identifiant_acheteur = None,
                 date_transaction = "",
                 date_liv_rec = ""):
        self.metal = metal
        self.quantite = quantité
        self.prix = prix
        self.identifiant_vendeur = identifiant_vendeur
        self.identifiant_acheteur = identifiant_acheteur
        self.date_transaction = date_transaction
        self.date_liv_rec = date_liv_rec

transactions = []
for _ in range(10000):
    transactions.append(Transaction())
```

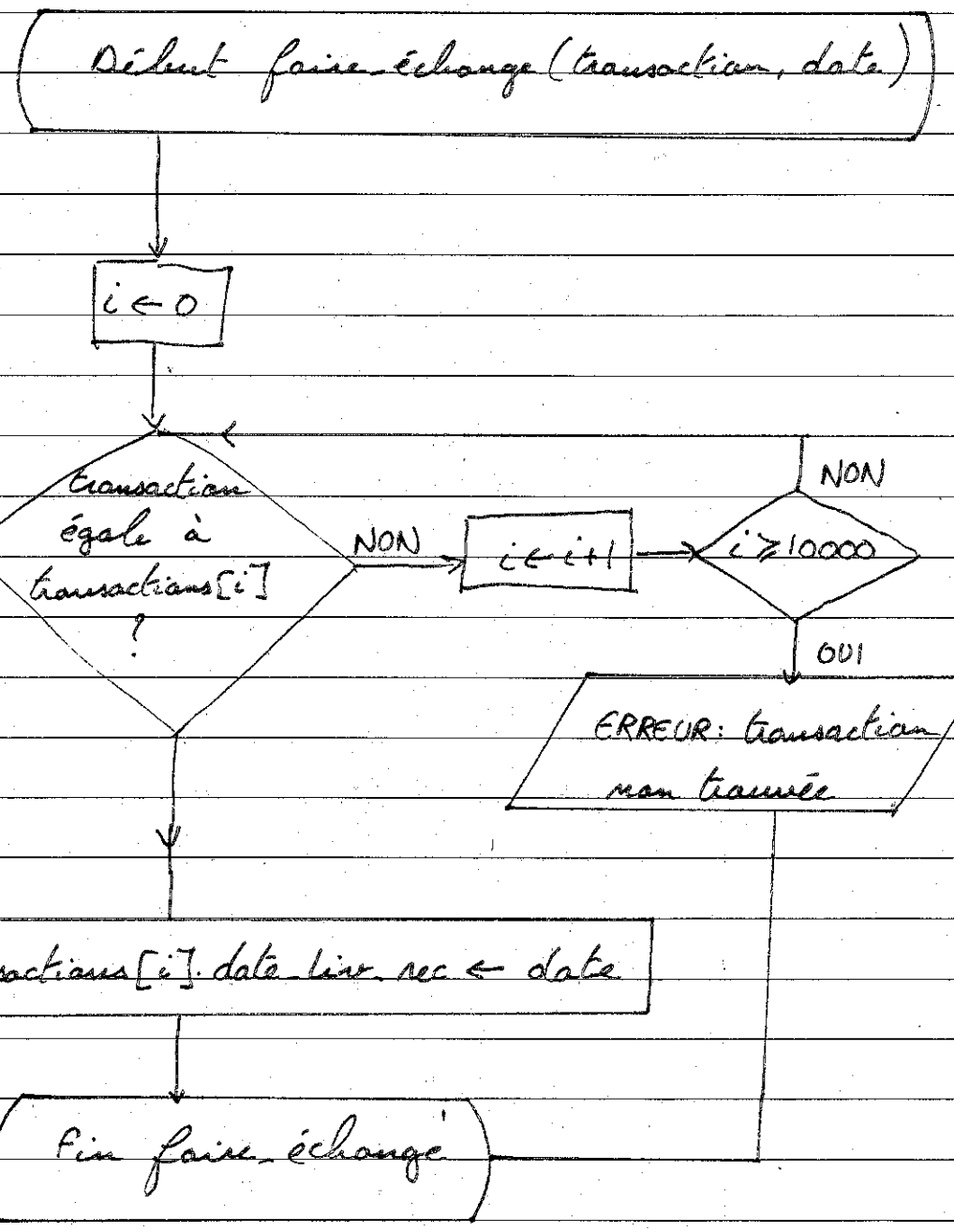
(Lignes verticales pour expliciter la tabulation)

Nombre intercalaire(s) 8

Question 8 /

```
def faire_transaction (metal, quantité, prix, identifiant_vendeur,
                       identifiant_acheteur, date_transaction,
                       date_liv_rec) :
    global transactions
    for i in range (10 000) :
        if transactions[i].metal not in (gold, platine, argent,
                                         palladium) :
            transactions[i] = Transaction (metal, quantite,
                                           prix, identifiant_vendeu
                                           identifiant_acheteur,
                                           date_transaction,
                                           date_liv_rec)
            return
    return "Tableau des transactions plein"
```

question 9 /

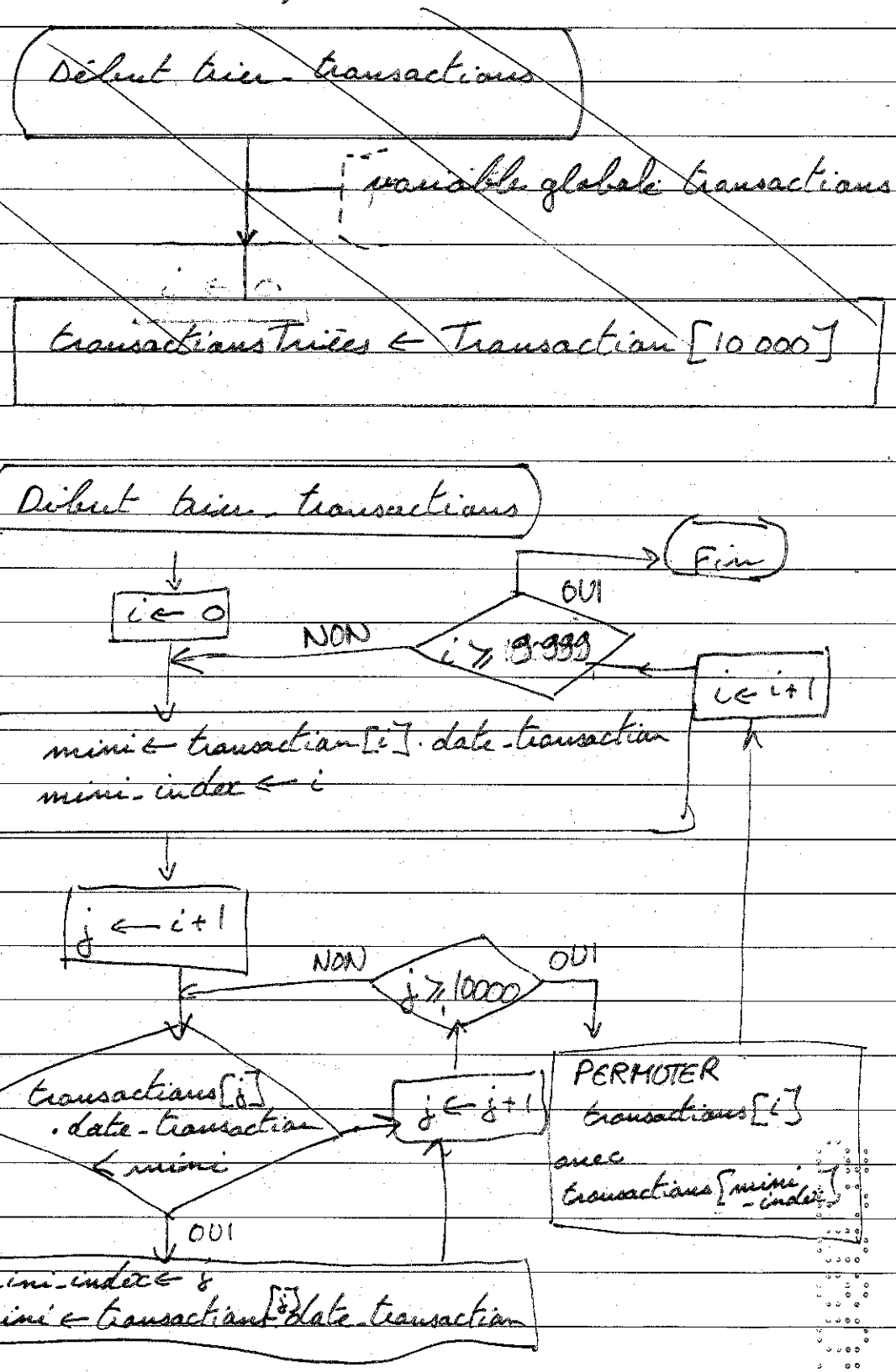

Début faire_échange (transaction, date)
i ← 0
transaction égale à transactions[i] ?
NON
i ← i+1
i ≥ 10000
NON
OUI
ERREUR: transaction non trouvée
transactions[i].date_livraison ← date
Fin faire_échange

Nombre d'intercalaire(s) 8

question 10/

```
def faire_echange(transaction, date):
    global transactions
    for i in range(10 000):
        if transactions[i] == transaction
            transactions[i].date_lire_sec = date
            return
    return "Transaction non trouvée"
```

question 11 / On effectue un "Tri à bulle"



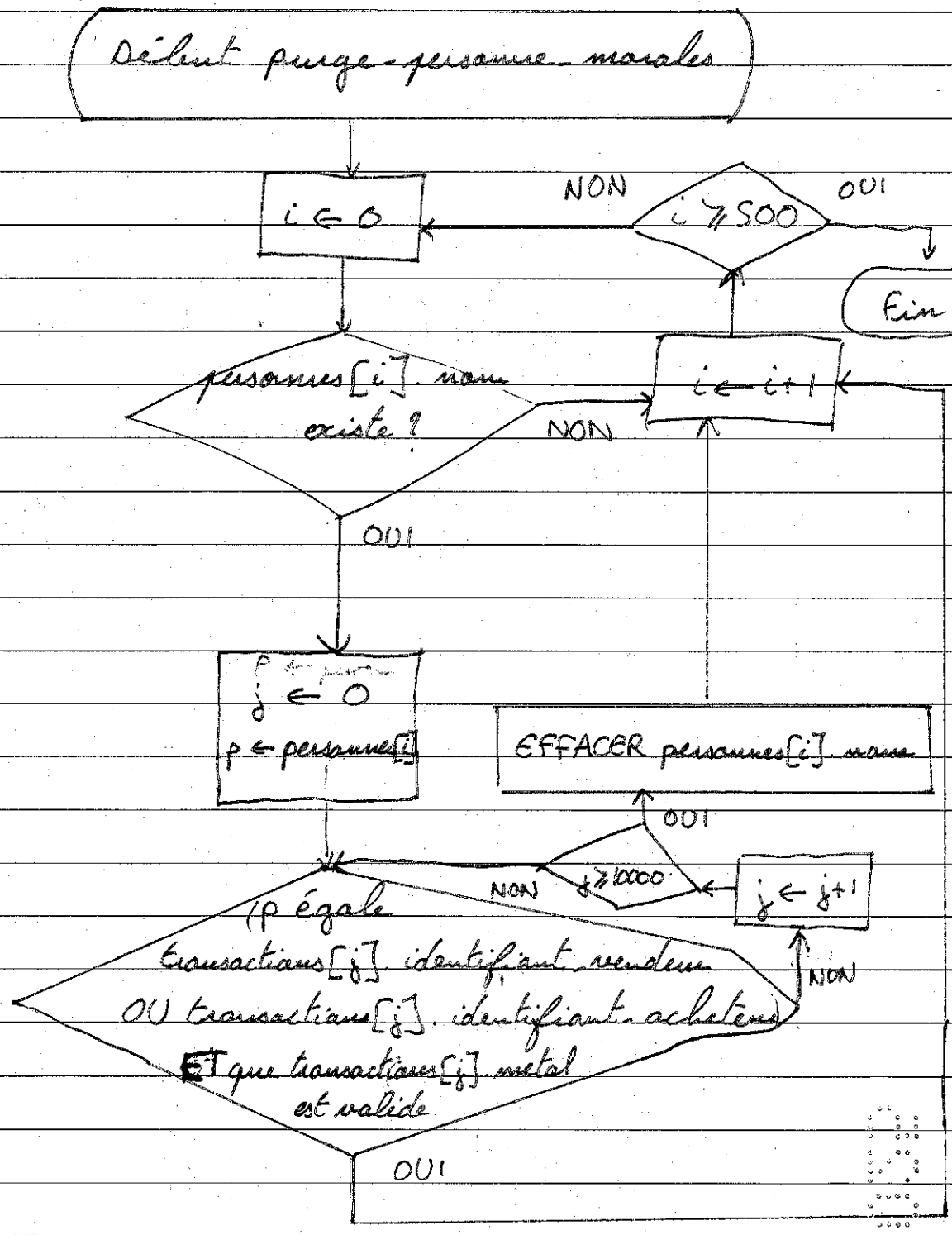
Nombre d'intercalaire(s) 8

question 12 /

```
def trier_transactions():
    global transactions

    for i in range(9 999):
        mini = transactions[i].date_transaction
        mini_index = i
        for j in range(i+1, 10 000):
            if transactions[j].date_transaction < mini
                mini_index = j
                mini = transactions[j].date_transaction
        transactions[i], transactions[mini_index]
            = transactions[mini_index], transactions[i
```

question 13 /


Début purge_personne_morales
i ← 0
NON
i ≠ 500
OUI
Fin
personnes[i].nom existe ?
NON
i ← i+1
OUI
j ← 0
p ← personnes[i]
EFFACER personnes[i].nom
OUI
NON
j ≠ 10000
j ← j+1
(p égale transactions[j].identifiant_vendeur OU transactions[j].identifiant_acheteur) ET que transactions[j].metal est valide
NON
OUI

Nombre d'intercalaire(s) 8

question 14 /

```
def purge_personne_morales():
    global personnes
    for i in range(500):
        purge = True
        if personnes[i] != None:
            p = personnes[i]
            for j in range(10 000):
                if (p in (transactions[j].identifiant_vendeur
                          transactions[j].identifiant_acheteur)
                    and (transactions[j].metal in (gold,
                                                   platine,
                                                   argent,
                                                   palladium)):
                    purge = False
                    break
            if purge:
                personnes[i].nom = None
```

question 15 /

Début restructurer - tableau - personnes - morales

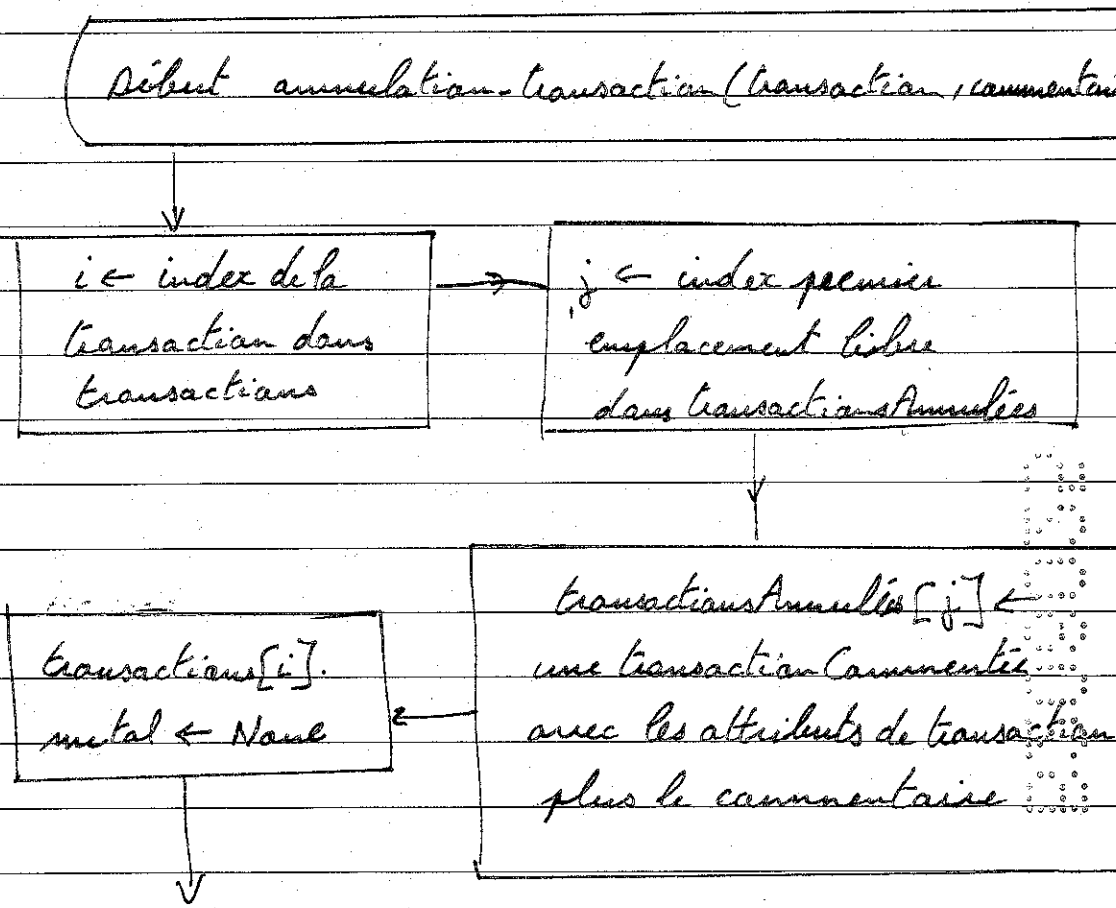
Nombre intercalaire(s) 8

question 16 /

question 17 /

je créerai un tableau "transactionsAnnulées" de 10 000 entrées de type TransactionCommentée défini comme suit :

Type TransactionCommentée :
Caractère : métal
Réel : quantité
Réel : prix
Entier : identifiant_vendeur
Entier : identifiant_acheteur
date : date_transaction
date : date_lim_rec
Chaîne de caractères : commentaire



Nombre intercalaire(s) 8

question 18 /

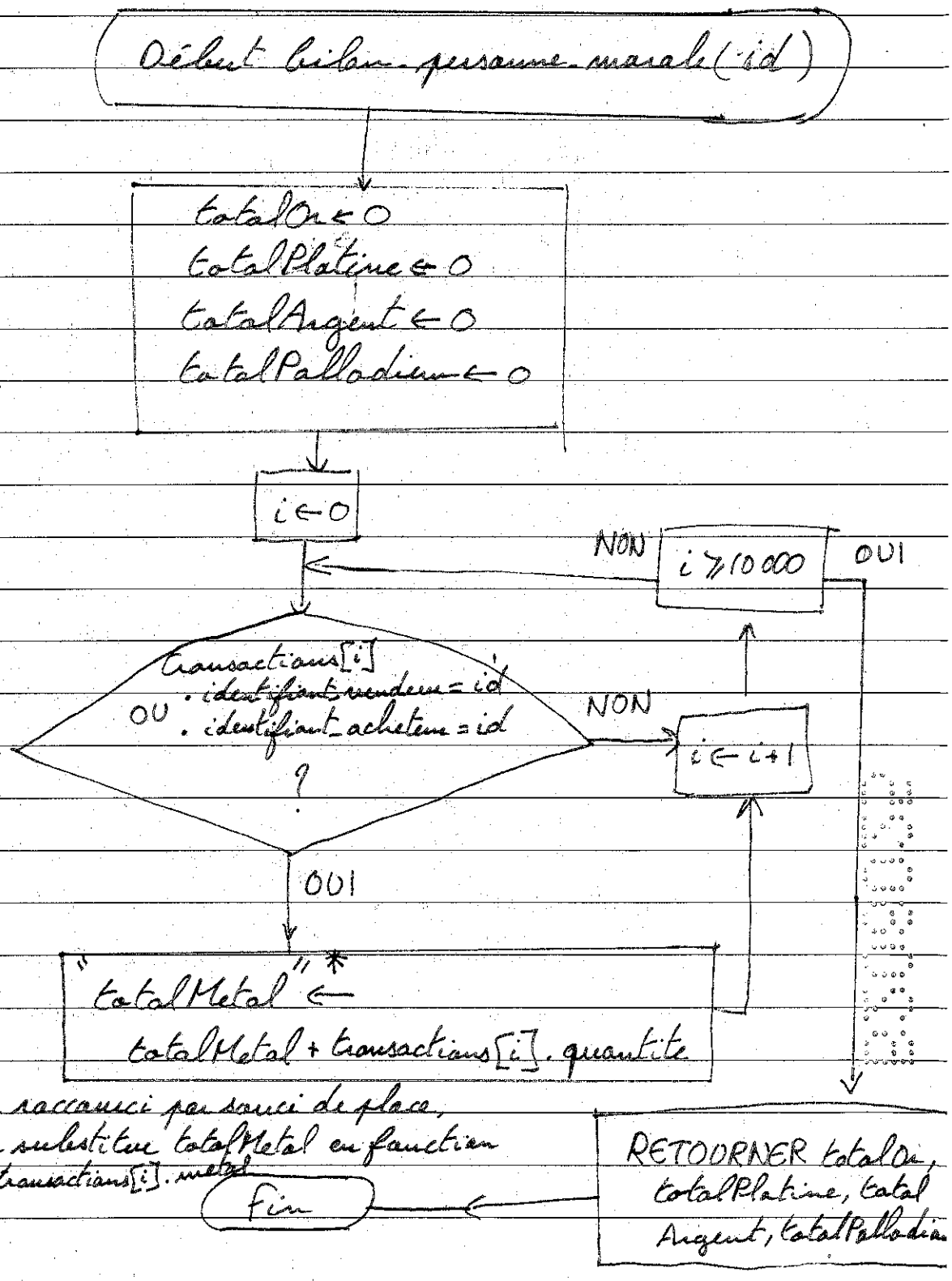
```
def annulation_transaction(transaction, commentaire):
    global transactions
    global transactionsAnnulées
    index_transaction = -1
    for i in range(10 000):
        if transactions[i] == transaction
            index_transaction = i
    if index_transaction == -1:
        break
        print("transaction non trouvée")
        return
    index_emplacement = -1
    for j in range(10 000):
        if transactionsAnnulées[j].metal not in
            (gold, platine, argent, palladium):
            index_emplacement = j
            break
    if index_emplacement == -1:
        print("Plus de place dans le tableau des
                annulations")
        return
    a = transactions[index_transaction]
    transactionsAnnulées[index_emplacement]
        = TransactionCommentée(a.metal, a.quantite,
                a.prix, a.identifiant_vendeur,
                a.identifiant_acheteur,
                a.date_transaction,
                a.date_livraison,
                commentaire)
```

(Le code reprends directement à une indentation du "def

transactions[indice_transaction].metal = Name

question 19 /

NB: On fait bien ici le total des mouvements en valeur absolue : achat ou vente sont traités de façon identique



Question 20 /

```
def bilan_pesanne_morale(id):

    totalOr, totalPlatine, totalArgent, totalPalladium = 0,0,0,0

    for t in transactions:
        if id in (t.identifiant_vendeur, t.identifiant_acheteur):
            if t.metal == gold:
                totalOr += t.quantite
            elif t.metal == platine:
                totalPlatine += t.quantite
            elif t.metal == argent:
                totalArgent += t.quantite
            elif t.metal == palladium:
                totalPalladium += t.quantite

    return (totalOr, totalPlatine, totalArgent, totalPalladium)
```