Rapport TP2 :

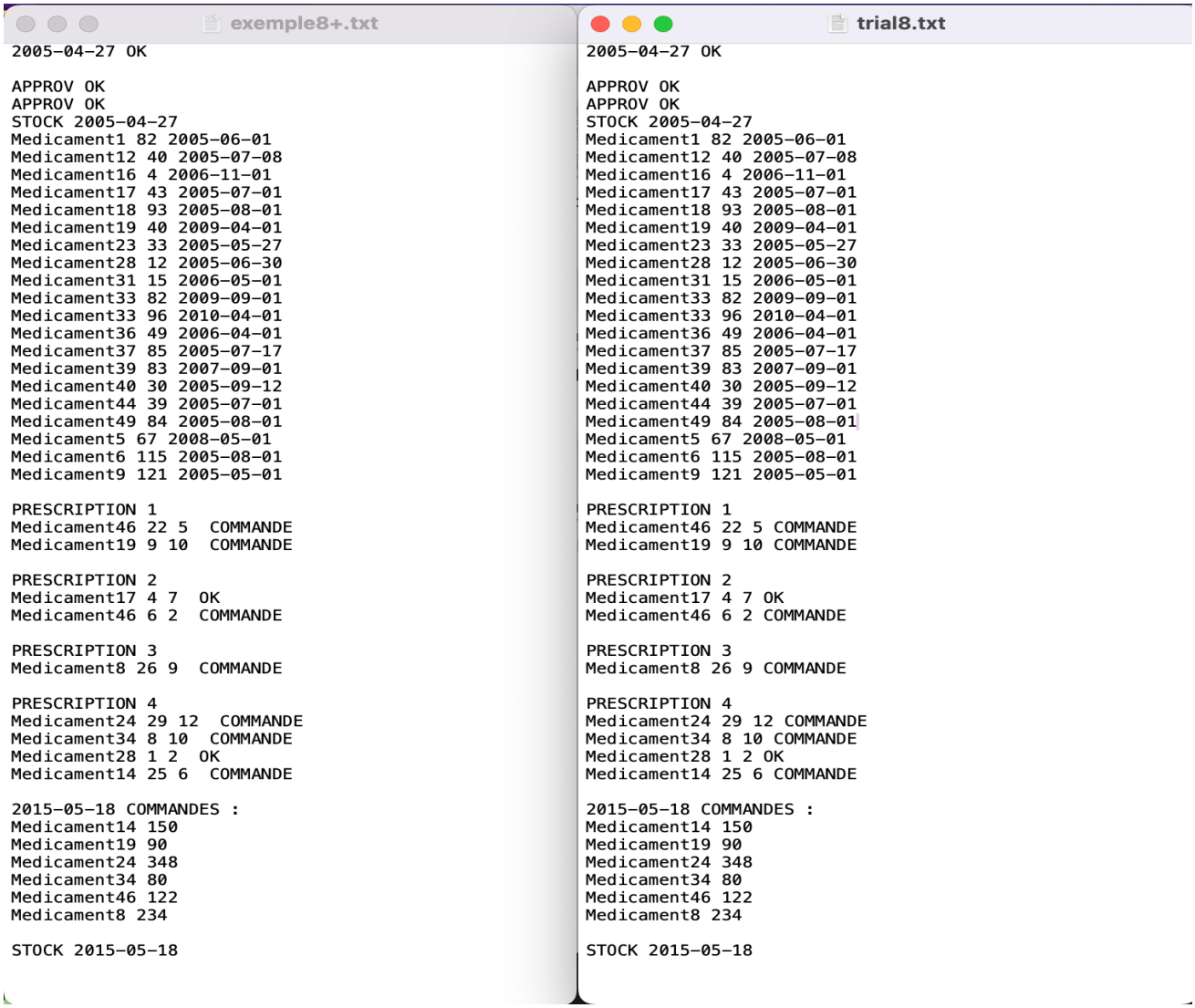
Reem Habib (20257644), Tai Foster (20267450) et Jules Lemee (20319400)

Autoévaluation :

Notre code marche super bien, tous les fichiers d'inputs donnés en exemple ont été run puis nos fichiers d'outputs sont tous identiques à ceux qui nous ont été fournis.

Un extra effort a aussi été fait pour afficher les médicaments dans le même ordre que le vôtre.

Vous trouverez joint-ci-dessous un exemple de comparaison entre votre fichier d'output (à gauche) exemple8+.txt et le nôtre (à droite) trial8.txt



exemple8+.txt

```
2005-04-27 OK

APPROV OK
APPROV OK
STOCK 2005-04-27
Medicament1 82 2005-06-01
Medicament12 40 2005-07-08
Medicament16 4 2006-11-01
Medicament17 43 2005-07-01
Medicament18 93 2005-08-01
Medicament19 40 2009-04-01
Medicament23 33 2005-05-27
Medicament28 12 2005-06-30
Medicament31 15 2006-05-01
Medicament33 82 2009-09-01
Medicament33 96 2010-04-01
Medicament36 49 2006-04-01
Medicament37 85 2005-07-17
Medicament39 83 2007-09-01
Medicament40 30 2005-09-12
Medicament44 39 2005-07-01
Medicament49 84 2005-08-01
Medicament5 67 2008-05-01
Medicament6 115 2005-08-01
Medicament9 121 2005-05-01

PRESCRIPTION 1
Medicament46 22 5  COMMANDE
Medicament19 9 10  COMMANDE

PRESCRIPTION 2
Medicament17 4 7  OK
Medicament46 6 2  COMMANDE

PRESCRIPTION 3
Medicament8 26 9  COMMANDE

PRESCRIPTION 4
Medicament24 29 12  COMMANDE
Medicament34 8 10  COMMANDE
Medicament28 1 2  OK
Medicament14 25 6  COMMANDE

2015-05-18 COMMANDES :
Medicament14 150
Medicament19 90
Medicament24 348
Medicament34 80
Medicament46 122
Medicament8 234

STOCK 2015-05-18
```

trial8.txt

```
2005-04-27 OK

APPROV OK
APPROV OK
STOCK 2005-04-27
Medicament1 82 2005-06-01
Medicament12 40 2005-07-08
Medicament16 4 2006-11-01
Medicament17 43 2005-07-01
Medicament18 93 2005-08-01
Medicament19 40 2009-04-01
Medicament23 33 2005-05-27
Medicament28 12 2005-06-30
Medicament31 15 2006-05-01
Medicament33 82 2009-09-01
Medicament33 96 2010-04-01
Medicament36 49 2006-04-01
Medicament37 85 2005-07-17
Medicament39 83 2007-09-01
Medicament40 30 2005-09-12
Medicament44 39 2005-07-01
Medicament49 84 2005-08-01
Medicament5 67 2008-05-01
Medicament6 115 2005-08-01
Medicament9 121 2005-05-01

PRESCRIPTION 1
Medicament46 22 5 COMMANDE
Medicament19 9 10 COMMANDE

PRESCRIPTION 2
Medicament17 4 7 OK
Medicament46 6 2 COMMANDE

PRESCRIPTION 3
Medicament8 26 9 COMMANDE

PRESCRIPTION 4
Medicament24 29 12 COMMANDE
Medicament34 8 10 COMMANDE
Medicament28 1 2 OK
Medicament14 25 6 COMMANDE

2015-05-18 COMMANDES :
Medicament14 150
Medicament19 90
Medicament24 348
Medicament34 80
Medicament46 122
Medicament8 234

STOCK 2015-05-18
```

Analyse temporelle :

- **Pour Prescription :**

Variables utilisées :
n : nombre de différents médicaments dans une prescription
s : nombre de différents médicaments dans le stock
l : nombre de dates d'expiration différentes d'un médicament donné

<u>Pour 1 médicament dans une Prescription :</u>

- ◊ **On fait l'opération search for med in tree qui nous prend log(s) car l'arbre implémenté est un arbre AVL et qu'au pire des cas pour chercher le médicament en question dans notre stock on fait log(s)**

- ◊ **On fait aussi l'opération search in subtree qui concerne les dates d'expiration pour chaque médicament, dans le pire des cas on fera log(l)**

- ◊ **Donc l'opération de search pour un médicament nous prend :**
  log(s) + log(l)

<u>Pour tous les médicaments dans une Prescription :</u>

- ◊ **On effectue les opérations citées au-dessus pour 1 médicament n fois pour pouvoir couvrir tous les différents médicaments d'une commande**

- ◊ **D'où au pire des cas, ça nous prend O(n(log(s) + log(l)))**

- **Pour DATE :**

Variables utilisées :
c : le nombre de commandes à effectuer par date
s : nombre de différents médicaments dans le stock
l : nombre de dates d'expiration différentes d'un médicament donné

- ◊ **On effectue l'opération writeOrders qui prend un temps équivalent à O(c)**

- ◊ **On effectue aussi removeExpiredMedications pour enlever les médicaments expirés donc on fait au pire des cas O(s*l) car pour chacun des médicaments présents dans le stock on va parcourir le subTree qui lui est associé qui appartient les dates d'expiration**

- ◊ **Donc la fonction update Date prendra un temps équivalent à O(c + s*l) au pire des cas**

- **Pour APPROV :**

Variables utilisées :
s : nombre de différents médicaments dans le stock
p : nombre de différents médicaments dans Approv

- ◊ **On effectuera l'opération search for med in tree qui nous prendra au pire des cas O(log(s)) car on cherche le médicament dans un arbre AVL**

- ◊ **On effectue aussi l'opération insert in stock qui nous prend au pire des cas O(log(s))**

- ◊ **Ainsi pour tous les médicaments présents dans Approv, on fera les opérations d'avant p fois d'où pire des cas on aura une complexité temporelle égale à O(p(log(s)))**

Prière de noter que m et k n’ont pas été utilisés car dans notre implémentation ces variables n’impactent pas vraiment la complexité temporelle de Prescription, Date ou de Approv

Si m (nombre d'items dans la prescription) correspond bien à doses * répétitions comme expliqué dans le discord, alors ça n'a aucun impact sur l'efficacité du programme car que le nombre d'items soit 10 ou 100000, ce nombre est utilisé seulement comme une condition pour voir si on a trouvé le bon noeud et pour soustraire cette quantité du stock.