



FACULTÉ DES SCIENCES ET TECHNIQUES DE SAINT JÉRÔME

SUJETS D'EXAMEN LMD

<i>Cycle</i>	LICENCE
<i>Série</i>	I2 – Architecture des machines et des systèmes
<i>Années</i>	2005-

Attention : un sujet peut comporter plusieurs pages

© 2005 - Université Paul Cézanne Aix-Marseille 3 - FST - BU

Toute reproduction, totale ou partielle, de ce document, est interdite sans le consentement express de son auteur

Université Paul Cézanne (Aix-Marseille III)

Licence Math-Info 1^e année - I2 - Examen de juin 2007 (2^e session)
Durée : 1h30 - Aucun document autorisé - Calculatrice interdite

Exercice 1 : Circuits logiques (7 points)

Le but final de l'exercice est de réaliser un circuit avec mémoire qui indique le nombre d'objets contenus dans une boîte au cours du temps. Cette boîte peut contenir de 0 à 3 objets. Un fournisseur peut de temps à autre décider d'ajouter un objet dans la boîte. Un consommateur peut de temps à autre décider de retirer un objet de la boîte.

Dans un premier temps, nous allons construire un circuit C avec 4 bits d'entrée n_1 , n_0 , f et c et 2 bits de sortie n'_1 et n'_0 . Les entrées n_1 et n_0 codent en représentation binaire le nombre d'objets contenus actuellement dans la boîte ($n_1 = 1$ et $n_0 = 0$ signifie qu'il y a deux objets). $f = 1$ signifie que le fournisseur veut ajouter un objet tandis que $f = 0$ signifie qu'il ne fait rien. $c = 1$ signifie que le consommateur veut retirer un objet et $c = 0$ signifie qu'il ne fait rien. Les sorties n'_1 et n'_0 codent le nombre d'objets que contiendra la boîte après que le fournisseur et/ou le consommateur aient modifié le nombre d'objets. Remarque : si $f = c$, le nombre d'objets ne change pas; si $f = 1$ et $c = 0$, il y aura un objet de plus (s'il n'y en avait pas trois); si $f = 0$ et $c = 1$, il y aura un objet de moins (s'il y en avait au moins un).

1.1 *Ecrire la table de vérité de la fonction logique qui indique les valeurs des sorties n'_1 et n'_0 en fonction des valeurs des entrées n_1 , n_0 , f et c .*

1.2 *En utilisant la méthode des tables de Karnaugh, écrire les fonctions booléennes correspondant aux valeurs de sorties de n'_1 et n'_0 .*

1.3 *Dessiner le circuit C permettant d'obtenir n'_1 et n'_0 à partir de n_1 , n_0 , f et c . Pour ceci, on n'a le droit d'utiliser que la porte logique unaire NON et les portes logiques OU et ET.*

1.4 Maintenant, on veut réaliser le circuit avec mémoire qui indique le nombre d'objets contenus dans la boîte au cours du temps. Ce circuit possède une entrée d'horloge nommée CK, trois entrées f , c et CLR et deux sorties n_1 et n_0 . Les sorties n_1 et n_0 indiquent le nombre actuel de jetons dans la boîte. Les entrées f et c ont le même sens qu'auparavant. Lorsque l'entrée CLR est à 0, la boîte est vidée, c'est-à-dire, les sorties n_1 et n_0 sont à 0. Lorsque CLR est à 1, à chaque impulsion d'horloge les sorties sont modifiées en fonction de f , de c et des précédentes valeurs de n_1 et n_0 .

Dessiner le circuit décrit ci-dessus. Pour ceci, on n'utilisera que le circuit C (défini dans la question 1.3), des bascules D et des portes logiques NON, OU ou ET. Dans cette question, le circuit C est considéré comme une boîte noire, on le dessinera comme un rectangle (à quatre entrées et deux sorties) et on ne redessinera donc pas les portes logiques qui le composent.

Exercice 2 : Représentation des nombres et codage en mémoire (3 points)

2.1 *Donnez la représentation binaire et la représentation hexadécimale du nombre 189 (représenté en notation décimale).*

2.2 *Donnez la valeur en représentation décimale de l'entier relatif dont le codage en binaire tient sur un octet et dont la représentation hexadécimale est AB.*

2.3 Le code morse permet notamment de coder les 26 lettres de l'alphabet par une suite de 1 à 4 traits (-) ou points (.). Par exemple, A se code .-, B se code -... , etc. *Combien faut-il de bits pour coder une lettre ? Si on ne tient pas compte des espaces entre les mots et de la ponctuation, combien faut-il d'octets pour coder un message contenant 81 lettres ?*

Exercice 3 : Langage machine (4 points)

Dans cet exercice, on se place dans le cadre du microprocesseur MP0 et du langage machine LM0 décrit en cours. On rappelle que MP0 est un microprocesseur possédant deux registres d'adresses 8 bits A0 et A1 et deux registres de données 8 bits D0 et D1. LM0 possède notamment les instructions de transfert d'octet MOVE, d'addition ADD, de soustraction SUB. Ces instructions ont deux opérands. Le premier est l'opérande source et le deuxième est l'opérande destination, qui sera modifié grâce à l'opérande source. LM0 a 3 types d'adressage : immédiat (on préfixe l'opérande avec #), direct (l'opérande est laissé tel quel) et indirect (l'opérande, qui est toujours un registre d'adresse, est mis entre parenthèses).

3.1 *On considère que la mémoire contient la suite de quatre valeurs 1, 3, 6 et 8 à partir de l'adresse 50. On exécute la série d'instructions suivante stockée à partir de l'adresse 100 :*

```
100 : MOVE #53, A0
102 : CMP #9, (A0)
104 : JEQ #112
106 : MOVE #0, (A0)
108 : SUB #1, A0
110 : JMP #102
112 : ADD #1, (A0)
```

Quelles cases de la mémoire seront-elles modifiées et quelles valeurs contiendront-elles après l'exécution de ces instructions? Mêmes questions s'il y avait eu à la place à partir de l'adresse 50 les valeurs 1, 3, 6 et 9? les valeurs 1, 9, 9 et 9? les valeurs 9, 9, 9 et 9? On rappelle que CMP est l'instruction de comparaison. JEQ est l'instruction conditionnelle de saut si la première valeur de l'instruction CMP précédente est égale à sa deuxième valeur.

3.2 *On exécute la série d'instructions suivante stockée à partir de l'adresse 100 :*

```
100 : MOVE #53, A0
102 : ADD #1, A0
104 : MOVE A0, (A0)
106 : ADD A0, A0
108 : JMP #102
```

Quel effet aura l'exécution de ces instructions? Expliquez pourquoi. On rappelle que chaque instruction est codée sur deux octets et que le deuxième octet contient un nombre apparaissant dans un des deux opérands.

30 JUIN 2007

3/3

Exercice 4 : Utilisation d'Unix (6 points)

4.1 On se trouve dans un répertoire ne contenant que les fichiers `bourgeois.txt` et `gentilhomme.txt`. Le fichier `bourgeois.txt` contient l'unique ligne de texte `Belle Marquise, vos beaux yeux et le fichier gentilhomme.txt` contient l'unique ligne de texte `Me font mourir d'amour`. Indiquez les effets du lancement des commandes Unix suivantes à partir d'un shell. Signalez les erreurs éventuelles. Chaque question est indépendante : on fera comme si la commande précédente n'avait pas été exécutée.

- a) `$ cat bourgeois.txt`
- b) `$ cat < gentilhomme.txt`
- c) `$ cat moliere.txt`
- d) `$ cat bourgeois.txt gentilhomme.txt`
- e) `$ cat bourgeois.txt > gentilhomme.txt`
- f) `$ cat bourgeois.txt >> gentilhomme.txt`
- g) `$ cat bourgeois.txt bourgeois.txt gentilhomme.txt | head -n2`
- h) `$ cat bourgeois.txt | cat`
- i) `$ cat | cat > gentilhomme.txt`
- j) `$ cat`
- k) `$ echo bourgeois.txt >> bourgeois.txt`
- l) `$ ls moliere.txt >> moliere.txt`

4.3 Indiquez les commandes permettant d'effectuer les opérations suivantes, sachant que le fichier `document.txt` et le répertoire `Rep` (ainsi que d'autres fichiers et répertoires) se trouvent dans le répertoire courant :

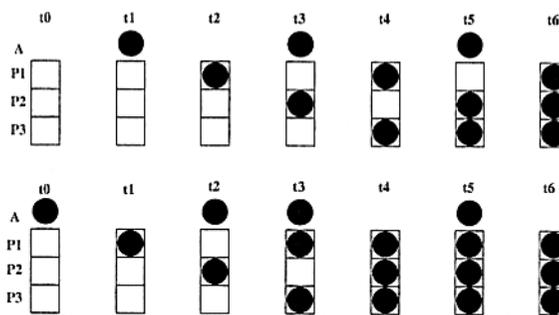
- a) afficher la liste de tous les fichiers du répertoire `Rep` dont le nom contient la suite de caractères `'xyz'` (sans se déplacer dans le répertoire `Rep`).
- b) afficher la troisième ligne du fichier `document.txt`.
- c) créer le fichier `fin_saisie.txt` constitué de la dernière ligne d'un texte entré au clavier.
- d) créer le fichier `liste` contenant la liste des fichiers et répertoires contenus dans `Rep`.
- e) afficher les trois premiers fichiers ou répertoires contenus dans le répertoire courant.
- f) rajouter à la fin du fichier `document.txt` la première ligne de ce même fichier.

Université Paul Cézanne (Aix-Marseille III)

Licence Math-Info 1ère année - I2 - Examen de mai 2007
 Durée : 1h30 - Aucun document autorisé - Calculatrice interdite

Exercice 1 : Circuits logiques (7 points)

Le but final de l'exercice est de réaliser un circuit avec mémoire qui indique l'état global d'une colonne de jetons qu'on complète au cours du temps. La figure ci-dessous indique deux exemples de l'évolution d'une colonne au cours du temps, pendant 6 unités de temps. La colonne peut contenir 3 jetons au maximum, dont les positions sont P1, P2 ou P3. On remplit cette colonne en ajoutant quand on veut un jeton par le haut, ce qu'indique la position A. Après qu'un jeton soit ajouté, il descend d'un cran dans la colonne à chaque unité de temps, jusqu'à toucher le sol (position P3) ou un jeton qui ne peut plus descendre (car ce dernier est sur le sol ou sur un jeton qui ne peut descendre). L'exemple du haut de la figure indique une succession d'états de la colonne où on ajoute un nouveau jeton au temps t1, au temps t3 et au temps t5. Le premier jeton se trouve en position A au temps t1, en position P1 au temps t2, en position P2 au temps t3 et en position P3 à partir du temps t4. Dans l'exemple du bas de la figure, le 3e jeton est ajouté au temps t3 (en position A) et reste bloqué en position P1 à partir du temps t4 car le 2e jeton (ajouté au temps t2) reste bloqué en position P2 à partir du temps t4. Au temps t5, on essaie d'ajouter un 4e jeton mais il n'y a plus de place donc rien ne change dans la colonne.



Dans un premier temps, nous allons construire un circuit C avec 4 bits d'entrée A , P_1 , P_2 et P_3 et 3 bits de sortie P'_1 , P'_2 et P'_3 . Ce circuit indique ce que sera l'état global d'une colonne au temps suivant, c'est-à-dire, le contenu de chacune des trois positions de la colonne, en fonction de ce que contenait précédemment la colonne en ses trois positions et du fait qu'on ajoute ou non un nouveau jeton. Quand un bit est à 1, cela signifie qu'il y a un jeton à la position qu'il indique. Ex: si on a en entrée $A = 1$, $P_1 = 0$, $P_2 = 1$ et $P_3 = 0$ (ce qui correspond à l'état au temps t3 de la colonne de l'exemple du haut de la figure) alors on a en sortie $P'_1 = 1$, $P'_2 = 0$ et $P'_3 = 1$ (contenu de la colonne au temps t4); si on a en entrée $A = 1$, $P_1 = 1$, $P_2 = 1$ et $P_3 = 1$ (ce qui correspond à l'état de la colonne de l'exemple du bas de la figure au temps t5) alors on a en sortie $P'_1 = 1$, $P'_2 = 1$ et $P'_3 = 1$ (contenu de la colonne au temps t6).

1.1 Ecrire la table de vérité de la fonction logique qui indique les valeurs des sorties P'_1 , P'_2 et P'_3 en fonction des valeurs d'entrées A , P_1 , P_2 et P_3 .

1.2 En utilisant la méthode des tables de Karnaugh, écrire les fonctions booléennes correspondant aux valeurs de sorties de P'_1 , P'_2 et P'_3 .

1.3 Dessiner le circuit C permettant d'obtenir P'_1 , P'_2 et P'_3 à partir de A , P_1 , P_2 et P_3 . Pour ceci, on n'a le droit d'utiliser que la porte logique unaire NON et les portes logiques OU et ET.

30 MAI 2007

2/3

1.4 Maintenant, on veut réaliser le circuit à mémoire simulant une succession d'états d'une colonne de jetons. Ce circuit possède une entrée d'horloge nommée CK, deux entrées a et CLR et trois sorties p_1 , p_2 et p_3 . Les sorties p_1 , p_2 et p_3 indiquent l'état de la colonne de jetons. L'entrée a indique si on ajoute un jeton ou non. Lorsque l'entrée CLR est à 0, la colonne est vidée, c'est-à-dire, les sorties p_1 , p_2 et p_3 sont à 0. Lorsque CLR est à 1, à chaque impulsion d'horloge les sorties sont modifiées en fonction de a et des précédentes valeurs de p_1 , p_2 et p_3 .

Dessiner le circuit décrit ci-dessus. Pour ceci, on n'utilisera que le circuit C (défini dans la question 1.3), des bascules D et des portes logiques NON, OU ou ET. Dans cette question, le circuit C est considéré comme une boîte noire, on le dessinera comme un rectangle (à quatre entrées et trois sorties) et on ne redessinera donc pas les portes logiques qui le composent.

Exercice 2: Représentation des nombres et codage en mémoire (3 points)

2.1 Donnez la représentation binaire et la représentation hexadécimale du nombre 123 (représenté en notation décimale).

2.2 Donnez la valeur en représentation décimale de l'entier relatif codé sur un octet dont la représentation hexadécimale est E3.

2.3 Un texte ASCII contient une suite de 500 caractères. Combien d'octets sont utilisés pour stocker ce fichier? Si ces caractères n'étaient que des chiffres (entre 0 et 9), combien de bits suffirait-il pour coder chaque chiffre? Combien d'octets pour stocker ces 500 chiffres?

Exercice 3: Langage machine (4 points)

Dans cet exercice, on se place dans le cadre du microprocesseur MP0 et du langage machine LM0 décrit en cours. On rappelle que MP0 est un microprocesseur possédant deux registres d'adresses 8 bits A0 et A1 et deux registres de données 8 bits D0 et D1. LM0 possède notamment les instructions de transfert d'octet MOVE, d'addition ADD, de soustraction SUB. Ces instructions ont deux opérandes. Le premier est l'opérande source et le deuxième est l'opérande destination, qui sera modifié grâce à l'opérande source. LM0 a 3 types d'adressage: immédiat (on préfixe l'opérande avec #), direct (l'opérande est laissé tel quel) et indirect (l'opérande, qui est toujours un registre d'adresse, est mis entre parenthèses).

3.1 On considère que la mémoire contient la suite de dix valeurs 56, 34, 5, 88, 61, 63, 16, 45, 72 et 36 à partir de l'adresse 50. On exécute la série d'instructions suivante stockée à partir de l'adresse 100:

```
100 : MOVE #50, D0
102 : MOVE D0, A1
104 : MOVE A1, A0
106 : ADD #1, A1
108 : CMP (A1), (A0)
110 : JGT #114
112 : MOVE A1, A0
114 : CMP #59, A1
116 : JGT #106
118 : MOVE D0, A1
120 : MOVE (A1), D1
122 : MOVE (A0), (A1)
124 : MOVE D1, (A0)
126 : ADD #1, D0
128 : CMP #59, D0
130 : JGT #102
```

30 MAI 2007

3/3

Quelles seront les 10 valeurs stockées à partir de l'adresse 50 après l'exécution de ces instructions? Expliquez pourquoi. On rappelle que CMP est l'instruction de comparaison. JGT est l'instruction conditionnelle de saut si la première valeur de l'instruction CMP précédente est strictement supérieure à sa deuxième valeur.

3.2 On exécute la série d'instructions suivante stockée à partir de l'adresse 60 :

```
60 : MOVE #65, A0
62 : ADD #2, (A0)
64 : ADD #2, A0
66 : ADD #2, (A0)
68 : MOVE #60, D0
70 : MOVE D0, 100
```

Quelle valeur y aura-t'il à l'adresse 100 après l'exécution de ces instructions? Expliquez pourquoi. On rappelle que chaque instruction est codée sur deux octets et que le deuxième octet contient un nombre apparaissant dans un des deux opérandes.

Exercice 4: Utilisation d'Unix (6 points)

On se trouve dans un répertoire contenant uniquement le fichier `grosminet.txt` et le répertoire `cage`, ce dernier ne contenant que le fichier `titi.txt`. Le fichier `grosminet.txt` contient l'unique ligne de texte `Miaou` et le fichier `titi.txt` contient l'unique ligne de texte `Cuicui`.

4.1 Indiquez les effets (ce qui est affiché, le ou les fichiers modifiés, etc) de l'exécution des commandes Unix suivantes à partir d'un shell. Signalez les erreurs éventuelles. Chaque question est indépendante: on fera comme si les commandes des questions précédentes n'avaient pas été exécutées.

- a) `$ ls`
- b) `$ ls grosminet.txt`
- c) `$ ls cage`
- d) `$ ls titi.txt`
- e) `$ ls *`
- f) `$ cat grosminet.txt`
- g) `$ cat *`
- h) `$ cat < grosminet.txt`
- i) `$ cat grosminet.txt > titi.txt`
- j) `$ cat > titi.txt`
- k) `$ cat`
- l) `$ echo Ouah >> grosminet.txt`
- m) `$ cat */* >> grosminet.txt`
- n) `$ cat grosminet.txt > grosminet.txt`
- o) `$ cat grosminet.txt >> grosminet.txt`
- p) `$ cat grosminet.txt cage/titi.txt grosminet.txt | head -n2`
- q) `$ cat grosminet.txt | cat > titi.txt`
- r) `$ cat grosminet.txt | cat >> grosminet.txt`
- s) `$ head -n2 | tail -n1`
- t) `$ ls chien.txt > chien.txt`

4.2 Indiquez la suite de commandes permettant de permuter les fichiers `grosminet.txt` et `titi.txt`.

4.3 Indiquez la suite de commandes permettant de créer le répertoire `maison` dans le répertoire courant et d'y déplacer `grosminet.txt` ainsi que le répertoire `cage`.

Université d'Aix-Marseille III

Licence Math-Info 1ère année - I2 - Examen de septembre 2006
Durée : 1h30 - Aucun document autorisé - Calculatrice interdite

Exercice 1 : Circuits logiques (7 points)

Nous allons construire des circuits déterminant des propriétés concernant une suite de 4 bits x_0, x_1, x_2 et x_3 , fournis en entrée du circuit. Ces circuits n'ont qu'une sortie s qui est à 1 si la propriété à déterminer est vraie et 0 si elle est fausse. Etant donné que ces 4 bits d'entrée forment une suite, on pourra simplifier leur expression en ne spécifiant que leur valeur dans l'ordre de la suite. Ainsi, si on dit qu'on fournit 0010 en entrée du circuit, cela signifie que $x_0 = 0, x_1 = 0, x_2 = 1$ et $x_3 = 0$.

1.1 Dans cette question, nous allons réaliser le circuit qui indique si la suite donnée en entrée contient un nombre de 1 supérieur ou égal à son nombre de 0. Cette propriété est donc vraie par exemple pour les entrées 1101, 1100, 1001 mais fausse par exemple pour les entrées 0000 et 0100.

Ecrire la table de vérité de la fonction logique qui indique la valeur de sortie s en fonction des valeurs des entrées x_0, x_1, x_2 et x_3 .

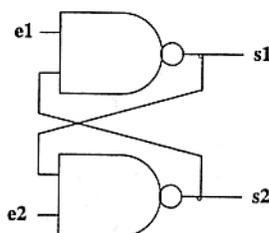
En utilisant la méthode des tables de Karnaugh, écrire la fonction booléenne correspondant à la valeur de sortie s .

Dessiner le circuit permettant d'obtenir s à partir de x_0, x_1, x_2 et x_3 . Pour ceci, on n'a le droit d'utiliser que la porte logique unaire NON et les portes logiques OU et ET.

1.2 Dans cette question, nous allons réaliser le circuit qui indique si la suite donnée en entrée contient au moins une fois deux 1 consécutifs (la sous-suite 11). Cette propriété est donc vraie par exemple pour les entrées 1101, 0110, 1111 mais fausse par exemple pour les entrées 0100, 0101 et 1001.

Définir ce circuit en répondant aux mêmes questions que précédemment : écrire la table de vérité, la table de Karnaugh et en déduire la fonction booléenne correspondante. (ne pas dessiner le circuit cette fois-ci)

1.3 Cette question est indépendante des questions 1.1 et 1.2.
Considérons le circuit ci-dessous.



Ecrire la table de vérité des sorties s_1 et s_2 en fonction des valeurs d'entrées e_1 et e_2 . Expliquer pourquoi certaines valeurs de sorties sont impossibles à déterminer uniquement en fonction des entrées e_1 et e_2 .

Pour quels changements simultanés des deux valeurs d'entrées e_1 et e_2 peut-on avoir une indécision quant aux valeurs de sorties s_1 et s_2 ? Expliquer pourquoi.

Exercice 2 : Représentation des nombres et codage en mémoire (3 points)

2.1 Donnez la représentation décimale et la représentation hexadécimale du nombre 1010111 (représenté en notation binaire).

2.2 Donnez la valeur en représentation décimale de l'entier relatif codé sur un octet dont la représentation binaire est 10101010 en complément à 2.

2.3 Combien de bits faut-il pour coder la date, sachant que le jour varie entre 1 et 31, le mois entre 1 et 12 et l'année entre 00 et 99? Combien d'octets?

Exercice 3 : Langage machine (4 points)

Dans cet exercice, on se place dans le cadre du microprocesseur MP0 et du langage machine LM0 décrit en cours. On rappelle que MP0 est un microprocesseur possédant deux registres d'adresses 8 bits A0 et A1 et deux registres de données 8 bits D0 et D1. LM0 possède notamment les instructions de transfert d'octet MOVE, d'addition ADD, de soustraction SUB. Ces instructions ont deux opérandes. Le premier est l'opérande source et le deuxième est l'opérande destination, qui sera modifié grâce à l'opérande source. LM0 a 3 types d'adressage: immédiat (on préfixe l'opérande avec #), direct (l'opérande est laissé tel quel) et indirect (l'opérande, qui est toujours un registre d'adresse, est mis entre parenthèses).

3.1 *Pour chaque instruction suivante, indiquez le registre de donnée, le registre d'adresse ou la case mémoire qui sera modifié ainsi que sa nouvelle valeur. Si l'instruction est incorrecte, indiquez-le et expliquez pourquoi.* On rappelle qu'une instruction est toujours mémorisée sur deux octets. Chaque instruction doit être considérée indépendamment des autres et les éventuelles modifications de la mémoire ou des registres des instructions précédentes ne seront pas prises en compte. Avant chaque exécution d'instruction, le registre D0 contient la valeur 5, le registre A0 contient la valeur 100 et la case mémoire d'adresse 100 contient la valeur 13.

- a) MOVE #100, D0
- b) MOVE A0, #100
- c) ADD 100, D0
- d) MOVE D0, 100
- e) MOVE 100, 101
- f) MOVE (A0), 101
- g) MOVE (A0), (A0)
- h) ADD (A0), (A0)

3.2 On considère que l'adresse 100 contient la valeur 6. On exécute la série d'instructions suivante stockée de l'adresse 200 à l'adresse 227:

```

200 : MOVE 100, D0
202 : SUB #1, D0
204 : MOVE 100, D1
206 : DIV D0, D1
208 : MUL D0, D1
210 : CMP 100, D1
212 : JEQ #222
214 : SUB #1, D0
216 : CMP #1, D0
218 : JEQ #226

```

220 : JMP #204
 222 : MOVE #0, 101
 224 : JMP #228
 226 : MOVE #1, 101
 228 : ...

Une fois arrivé à l'adresse 228, quelle valeur sera à l'adresse 101 ? Même question s'il y avait eu la valeur 7 (et non pas 6) à l'adresse 100 ? D'une façon générale, indiquez la valeur qu'on a à la fin à l'adresse 101 en fonction du type de valeur qu'on a à l'adresse 100. On rappelle que l'instruction JEQ est le saut conditionnel qui effectue un saut à l'adresse indiquée si les deux opérandes de l'instruction CMP la précédant ont la même valeur. On rappelle aussi que MUL est l'instruction de multiplication et que DIV est l'instruction de division entière.

Exercice 4 : Utilisation d'Unix (6 points)

4.1 On se trouve dans un répertoire ne contenant que les fichiers `deserteur.txt` et `vian.txt`. Le fichier `deserteur.txt` contient les 2 lignes de texte suivantes :

Monsieur le President
 Je vous fais une lettre

et le fichier `vian.txt` contient les 2 lignes de texte suivantes :

Que vous lirez peut-etre
 Si vous avez le temps.

Indiquez les effets du lancement des commandes Unix suivantes à partir d'un shell. Signalez les erreurs éventuelles. Chaque question est indépendante : on fera comme si la commande précédente n'avait pas été exécutée.

- a) `$ cat deserteur.txt`
- b) `$ cat < vian.txt`
- c) `$ cat boris.txt`
- d) `$ cat deserteur.txt vian.txt`
- e) `$ echo 'Je viens de recevoir' > deserteur.txt`
- f) `$ echo 'Mes papiers militaires' >> deserteur.txt`
- g) `$ cat vian.txt deserteur.txt | grep s`
- h) `$ echo 'Pour partir a la guerre' | cat`
- i) `$ cat | cat > deserteur.txt`
- j) `$ ls *.txt > deserteur.txt`
- k) `$ cat deserteur.txt >> vian.txt`
- l) `$ ls boris.txt > boris.txt`

4.2 Expliquer ce qui se passe lorsqu'on exécute les commandes suivantes :

- 1) `$ cat > machin`
- 2) `$ cat`
- 3) `$ head -2`
- 4) `$ head -2 > truc`
- 5) `$ cat chose >> chose`
- 6) `$ tail -5 | grep e`

Université d'Aix-Marseille III

Licence Math-Info 1ère année - I2 - Examen de juin 2006
Durée : 1h30 - Aucun document autorisé - Calculatrice interdite

Exercice 1 : Circuits logiques (7 points)

Le but final de l'exercice est de réaliser un circuit avec mémoire qui a deux fonctionnalités : l'incrémementation et la rotation circulaire.

Dans un premier temps, nous allons construire un circuit avec 4 bits d'entrée f , e_2 , e_1 et e_0 et 3 bits de sortie s_2 , s_1 et s_0 . Ce circuit a 2 fonctionnalités :

- si le bit d'entrée f est à 0, il fournit en sortie le successeur de l'entier codé en binaire par les bits e_2 , e_1 et e_0 . Par convention, le successeur de 111 est 000. e_2 code les "centaines", e_1 code les "dizaines" et e_0 code les "unités". L'entier binaire 011 est codé par $e_2 = 0$, $e_1 = 1$ et $e_0 = 1$. Les trois bits de sortie s_2 , s_1 et s_0 codent l'entier résultat de la même manière.
- si le bit d'entrée f est à 1, il fournit en sortie la rotation circulaire à droite des bits e_2 , e_1 et e_0 : on a $s_0 = e_1$, $s_1 = e_2$ et $s_2 = e_0$.

1.1 *Ecrire la table de vérité de la fonction logique qui indique les valeurs des sorties s_2 , s_1 et s_0 en fonction des valeurs des entrées f , e_2 , e_1 et e_0 .*

1.2 *En utilisant la méthode des tables de Karnaugh, écrire les fonctions booléennes correspondant aux valeurs de sorties de s_2 , s_1 et s_0 .*

1.3 *Dessiner le circuit permettant d'obtenir s_2 , s_1 et s_0 à partir de f , e_2 , e_1 et e_0 . Pour ceci, on n'a le droit d'utiliser que la porte logique unaire NON et les portes logiques OU et ET.*

1.4 Maintenant, on veut réaliser le circuit qui peut servir de compteur 3 bits ou de rotateur de bits. Il s'agit d'un circuit à mémoire qui a une entrée nommée CLR, une entrée d'horloge nommée CK, une entrée f qui sélectionne la fonctionnalité du circuit, et trois sorties s_2 , s_1 et s_0 qui indiquent les valeurs de sorties avec les mêmes conventions que les questions précédentes. Lorsque l'entrée CLR est à 0, les trois sorties sont à 0. Lorsque CLR est à 1, à chaque impulsion d'horloge le résultat en sortie est modifié en respectant la fonctionnalité sélectionnée par f .

Dessiner le circuit qui est décrit ci-dessus. Pour ceci, on n'utilisera que le circuit de la question 1.3, des bascules D et des portes logiques NON, OU ou ET. Dans cette question, le circuit de la question 1.3 est considéré comme une boîte noire, on le dessinera comme un rectangle (à quatre entrées et trois sorties) et on ne redessinera donc pas les portes logiques qui le composent.

Exercice 2 : Représentation des nombres et codage en mémoire (3 points)

2.1 Donnez la représentation binaire et la représentation hexadécimale du nombre 123 (représenté en notation décimale).

2.2 Donnez la valeur en représentation décimale de l'entier relatif codé sur un octet dont la représentation hexadécimale est E1.

2.3 Combien d'octets faut-il pour coder une image dont chaque point est soit blanc, soit noir, soit rouge, soit vert, et dont la résolution est 320×200 ?

Exercice 3 : Langage machine (4 points)

Dans cet exercice, on se place dans le cadre du microprocesseur MP0 et du langage machine LM0 décrit en cours. On rappelle que MP0 est un microprocesseur possédant deux registres d'adresses 8 bits A0 et A1 et deux registres de données 8 bits D0 et D1. LM0 possède notamment les instructions de transfert d'octet MOVE, d'addition ADD, de soustraction SUB. Ces instructions ont deux opérandes. Le premier est l'opérande source et le deuxième est l'opérande destination, qui sera modifié grâce à l'opérande source. LM0 a 3 types d'adressage: immédiat (on préfixe l'opérande avec #), direct (l'opérande est laissé tel quel) et indirect (l'opérande, qui est toujours un registre d'adresse, est mis entre parenthèses).

3.1 On considère que la mémoire contient la suite de dix valeurs 1, 2, 3, 6, 2, 1, 3, 2, 4, 2 à partir de l'adresse 50. On exécute la série d'instructions suivante stockée à partir de l'adresse 100 :

```
100 : MOVE #50, A0
102 : MOVE (A0), D0
104 : CMP #59, A0
106 : JEQ #116
108 : ADD #1, A0
110 : ADD (A0), D0
112 : MOVE D0, (A0)
114 : JMP #104
116 : ...
```

Quelles valeurs seront aux adresses 50 à 59 après l'exécution de ces instructions? Expliquer pourquoi. On rappelle que CMP est l'instruction de comparaison. JEQ est l'instruction conditionnel de saut si les deux valeurs comparées (par l'instruction CMP) sont égales.

3.2 On exécute la série d'instructions suivante stockée à partir de l'adresse 20 :

```
20 : MOVE #3, A0
22 : MOVE A0, (A0)
24 : ADD A0, A0
26 : CMP #100, A0
28 : JGT #22
```

Quel est l'effet de l'exécution de ces instructions? Expliquer pourquoi. On rappelle que l'instruction JGT est le saut conditionnel qui effectue un saut à l'adresse indiquée par son opérande si la valeur désignée par le premier opérande de l'instruction CMP (qui précède l'instruction JGT) est strictement supérieure à la valeur désignée par le deuxième opérande (de la même instruction CMP).

Exercice 4 : Utilisation d'Unix (6 points)

4.1 On se trouve dans un répertoire ne contenant que les fichiers f1.txt et f2.txt. Le fichier f1.txt contient l'unique ligne de texte PREMIER et le fichier f2.txt contient l'unique ligne de texte SECOND. Indiquez les effets de l'exécution des commandes Unix suivantes à partir d'un shell. Signalez les erreurs éventuelles. Chaque question est indépendante: on fera comme si la commande précédente n'avait pas été exécutée.

- a) \$ cat f1.txt
- b) \$ cat < f2.txt
- c) \$ cat f3.txt
- d) \$ cat f1.txt > f2.txt
- e) \$ cat f2.txt >> f1.txt
- f) \$ cat f1.txt f2.txt f1.txt f2.txt | head -n3
- g) \$ cat > f1.txt
- h) \$ cat
- i) \$ head -n1 > f3.txt

- j) `$ head -n2`
- k) `$ echo f1.txt >> f1.txt`
- l) `$ ls f3.txt > f3.txt`

4.2 On se trouve dans un répertoire contenant initialement l'unique répertoire `Rep`, qui contient lui-même uniquement un fichier `abc` et un répertoire `SousRep`, qui contient lui-même l'unique fichier `def`. Donner les commandes qui permettent successivement d'effectuer les opérations suivantes :

- déplacer le fichier `abc` dans le répertoire `SousRep` et le fichier `def` dans le répertoire `Rep`.
- se déplacer dans `SousRep`, puis, en restant dans `SousRep`, changer le nom du fichier `def` (qui est maintenant dans `Rep`) en `abc`,
- échanger le contenu des deux fichiers `abc` (celui dans `Rep` et celui dans `SousRep`).
- effacer le répertoire `Rep` ainsi que ce qu'il contient.

4.3 Indiquez les commandes permettant d'effectuer les opérations suivantes :

- a) En une seule ligne de commande, créer le fichier `deux.txt` contenant uniquement la deuxième ligne d'un texte entré au clavier.
- b) En une seule ligne de commande, afficher le nombre de lignes contenant le mot `diantre` présentes dans le fichier `saperlipopette.txt`.

Université d'Aix-Marseille III

Licence Math-Info 1ère année - I2 - Examen de juin 2005 (2e session)
Durée : 1h30 - Aucun document autorisé - Calculatrice interdite

Exercice 1 : Circuits logiques (7 points)

Nous allons construire des circuits permettant de comparer deux entiers de 2 bits. Ces circuits ont 4 entrées : x_1 , x_0 , y_1 et y_0 . Ils n'ont qu'une sortie s qui est à 1 si la comparaison effectuée est vraie et 0 si elle est fausse. Les deux bits d'entrée x_1 et x_0 permettent de coder le premier entier. x_1 code les "dizaines" et x_0 code les "unités". Ainsi, les entiers binaires possibles sont 00, 01, 10 et 11. L'entier 01 est codé par $x_1=0$ et $x_0=1$. Les deux bits d'entrée y_1 et y_0 permettent de coder le deuxième entier de la même manière que le premier.

1.1 Dans cette question, nous allons réaliser le circuit qui indique si le premier entier (codé sur 2 bits) est différent du second. La sortie s du circuit est donc à 0 uniquement si $x_1 = y_1$ et $x_0 = y_0$.

Écrire la table de vérité de la fonction logique qui indique la valeur de sortie s en fonction des valeurs des entrées x_1 , x_0 , y_1 et y_0 .

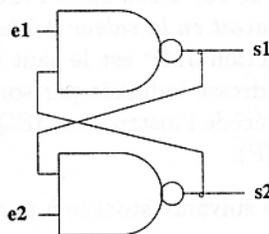
En utilisant la méthode des tables de Karnaugh, écrire la fonction booléenne correspondant à la valeur de sortie s .

Dessiner le circuit permettant d'obtenir s à partir de x_1 , x_0 , y_1 et y_0 . Pour ceci, on n'a le droit d'utiliser que la porte logique unaire NON et les portes logiques OU, ET et X-OU (OU eXclusif).

1.2 Dans cette question, nous allons réaliser le circuit qui indique si le premier entier (codé sur 2 bits) est supérieur ou égal au second. La sortie s du circuit est donc à 1 uniquement si $x_1 > y_1$ ou bien si $x_1 = y_1$ et $x_0 \geq y_0$.

Définir ce circuit en répondant aux mêmes questions que précédemment : écrire la table de vérité, la table de Karnaugh et en déduire la fonction booléenne correspondante. (ne pas dessiner le circuit cette fois-ci)

1.3 Cette question est indépendante des questions 1.1 et 1.2.
 Considérons le circuit ci-dessous.



Écrire la table de vérité des sorties s_1 et s_2 en fonction des valeurs d'entrées e_1 et e_2 . Expliquer pourquoi certaines valeurs de sorties sont impossibles à déterminer uniquement en fonction des entrées e_1 et e_2 .

Pour quels changements simultanés des deux valeurs d'entrées e_1 et e_2 peut-on avoir une indétermination quant aux valeurs de sorties s_1 et s_2 ? Expliquer pourquoi.

Exercice 2 : Codage en mémoire (2 points)

Le numéro INSEE (ou numéro de sécurité sociale) code un certain nombre de renseignements sur une personne, notamment 1 - le sexe (homme ou femme), 2 - les deux derniers chiffres de l'année de naissance (de 0 à 99), 3 - le mois de naissance (de 1 à 12) et 4 - le numéro de département de naissance (de 1 à 99).

Pour chacune de ces 4 informations, indiquez combien de bits sont nécessaires pour coder l'information. Si on voulait mémoriser ensemble ces 4 informations en utilisant le moins de mémoire possible, de combien d'octets aurait-on besoin au minimum ? (ne vous basez pas sur la méthode de codage utilisée en pratique pour votre numéro INSEE)

Exercice 3 : Langage machine (4 points)

Dans cet exercice, on se place dans le cadre du microprocesseur MP0 et du langage machine LM0 décrit en cours. On rappelle que MP0 est un microprocesseur possédant deux registres d'adresses 8 bits A0 et A1 et deux registres de données 8 bits D0 et D1. LM0 possède notamment les instructions de transfert d'octet MOVE, d'addition ADD, de soustraction SUB. Ces instructions ont deux opérandes. Le premier est l'opérande source et le deuxième est l'opérande destination, qui sera modifié grâce à l'opérande source. LM0 a 3 types d'adressage: immédiat (on préfixe l'opérande avec #), direct (l'opérande est laissé tel quel) et indirect (l'opérande, qui est toujours un registre d'adresse, est mis entre parenthèses).

3.1 On considère que la mémoire contient les six valeurs suivantes de l'adresse 100 à l'adresse 105: 4, 10, 8, 13, 23 et 6. On exécute la série d'instructions suivante stockée à partir de l'adresse 50 :

```
50 : MOVE 100, D0
52 : MOVE #101, A0
54 : ADD D0, A0
56 : MOVE (A0), D1
58 : MOVE A0, A1
60 : CMP #101, A0
62 : JEQ 72
64 : SUB #1, A1
66 : MOVE (A1), (A0)
68 : SUB #1, A0
70 : JMP 60
72 : MOVE D1, (A0)
```

Quelles valeurs seront aux adresses de 100 à 105 après l'exécution de ces instructions ? Expliquer pourquoi. Même question si à l'adresse 100, il y avait eu la valeur 2 à la place de la valeur 4. On rappelle que CMP est l'instruction de comparaison. L'instruction JMP est le saut inconditionnel et l'instruction JEQ est le saut conditionnel qui effectue un saut à l'adresse indiquée par son opérande si la valeur désignée par le premier opérande de l'instruction CMP (qui précède l'instruction JEQ) est égale à la valeur désignée par le deuxième opérande (de la même instruction CMP).

3.2 On exécute la série d'instructions suivante stockée à partir de l'adresse 200 :

```
200 : MOVE #220, A0
202 : MOVE #0, D0
204 : MOVE D0, (A0)
206 : SUB #1, A0
208 : ADD #1, D0
210 : CMP #20, D0
212 : JNE 204
```

Quel est l'effet de l'exécution de ces instructions ? Expliquer pourquoi. On rappelle que JNE est l'instruction conditionnel de saut si les deux valeurs comparées (par l'instruction CMP) sont différentes.

Exercice 4 : Utilisation d'Unix (7 points)

4.1 On se trouve dans un répertoire ne contenant que les fichiers `chanson.txt` et `d-automne.txt`. Le fichier `chanson.txt` contient les 3 lignes de texte suivantes :

```
Les sanglots longs
Des violons
De l'automne
```

et le fichier `d-automne.txt` contient les 3 lignes de texte suivantes :

```
Bercent mon coeur
D'une langueur
Monotone.
```

Indiquez les effets du lancement des commandes Unix suivantes à partir d'un shell. Signalez les erreurs éventuelles. Chaque question est indépendante : on fera comme si la commande précédente n'avait pas été exécutée.

- a) `$ cat chanson.txt`
- b) `$ cat < d-automne.txt`
- c) `$ cat verlaine.txt`
- d) `$ cat chanson.txt d-automne.txt`
- e) `$ echo 'Tout suffocant' > chanson.txt`
- f) `$ echo 'Et bleme, quand' d-automne.txt >> chanson.txt`
- g) `$ cat d-automne.txt chanson.txt | grep D`
- h) `$ echo 'Sonne 1 heure' | cat`
- i) `$ cat | cat > chanson.txt`
- j) `$ ls *.txt > chanson.txt`
- k) `$ cat chanson.txt >> d-automne.txt`
- l) `$ ls verlaine.txt > verlaine.txt`

4.2 On se trouve dans un répertoire `Rep` contenant initialement uniquement un répertoire `A`. Le répertoire `A` contient un certain nombre de fichiers mais aucun sous-répertoire.

Donner les commandes qui permettent successivement d'effectuer les opérations suivantes :

- dans le répertoire `Rep`, créer le répertoire `A2` contenant les mêmes fichiers que le répertoire `A`.
- supprimer le répertoire `A` (ainsi que son contenu).
- créer le fichier `liste` contenant la liste des fichiers du répertoire `A2` suivi d'une ligne contenant le nombre de fichiers de ce répertoire.

4.3 Expliquer ce qui se passe lorsqu'on exécute les commandes suivantes :

- 1) `$ cat > texte`
- 2) `$ cat`
- 3) `$ head -2`
- 4) `$ head -2 > texte`
- 5) `$ cat texte >> texte`

Université d'Aix-Marseille III

Licence Math-Info 1ère année - I2 - Examen de mai 2005
Durée : 1h30 - Aucun document autorisé - Calculatrice interdite

Exercice 1 : Circuits logiques (7 points)

Le but final de l'exercice est de réaliser un circuit avec mémoire qui fait office de compteur/décompteur.

Dans un premier temps, nous allons construire un circuit avec 4 bits d'entrées et 2 bits de sorties. Ce circuit a 4 fonctionnalités: à partir d'un entier codé sur 2 bits, il fournit en sortie soit l'entier zéro, soit le même entier qu'en entrée, soit son successeur, soit son prédécesseur. Deux bits d'entrée e_1 et e_0 permettent de coder l'entier. e_1 code les "dizaines" et e_0 code les "unités". Ainsi, les entiers binaires possibles sont 00, 01, 10 et 11. L'entier 01 est codé par $e_1=0$ et $e_0=1$. Les deux bits de sortie s_1 et s_0 codent l'entier résultat de la même manière. Les deux autres bits d'entrée f_1 et f_2 codent la fonctionnalité du circuit :

- Si $f_1=f_2=0$ alors l'entier de sortie est toujours 00, quel que soit l'entier en entrée.
- Si $f_1=0$ et $f_2=1$ alors l'entier de sortie est égal à l'entier d'entrée.
- Si $f_1=1$ et $f_2=0$, l'entier de sortie est le successeur de l'entier d'entrée: 01 succède à 00, 10 succède à 01, 11 succède à 10 et 00 succède à 11.
- Si $f_1=f_2=1$ alors l'entier de sortie est le prédécesseur de l'entier d'entrée. En particulier, 11 précède 00, 00 précède 01, etc.

1.1 *Ecrire la table de vérité de la fonction logique qui indique les valeurs des sorties s_1 et s_0 en fonction des valeurs des entrées f_1, f_2, e_1 et e_0 .*

1.2 *En utilisant la méthode des tables de Karnaugh, écrire les fonctions booléennes correspondant aux valeurs de sorties de s_1 et s_0 .*

1.3 *Dessiner le circuit permettant d'obtenir s_1 et s_0 à partir de f_1, f_2, e_1 et e_0 . Pour ceci, on n'a le droit d'utiliser que la porte logique unaire NON et les portes logiques binaires OU et ET.*

1.4 Maintenant, on veut réaliser le circuit qui peut servir de compteur 2 bits (incrémentement d'une unité à chaque cycle d'horloge) ou de décompteur 2 bits (décrémentement d'une unité à chaque cycle d'horloge), dont on peut figer la valeur de l'entier en sortie et qu'on peut réinitialiser à zéro.

Il s'agit d'un circuit à mémoire qui a une entrée nommée CLR, une entrée d'horloge nommée CK, deux entrées f_1 et f_2 qui sélectionnent la fonctionnalité du circuit comme précédemment, et 2 sorties s_1 et s_0 qui indiquent la valeur de l'entier avec les mêmes conventions que les questions précédentes. Lorsque l'entrée CLR est à 0, les deux sorties sont à 0. Lorsque CLR est à 1, à chaque impulsion d'horloge le nombre en sortie est modifié en fonction de la fonctionnalité codée par les entrées f_1 et f_2 .

Dessiner le circuit qui est décrit ci-dessus. Pour ceci, on n'utilisera que le circuit de la question **1.3**, des bascules D et des portes logiques NON, OU ou ET. Dans cette question, le circuit de la question **1.3** est considéré comme une boîte noire, on le dessinera comme un rectangle (à 4 entrées et 2 sorties) et on ne redessinera donc pas les portes logiques qui le composent.

Exercice 2 : Codage en mémoire (2 points)

On veut coder une position du jeu de go. Le plateau de jeu de go est une grille de 19 × 19 dont chacune des cases peut contenir (indépendamment des autres cases) une pierre blanche, une pierre noire ou pas de pierre.

Pour chacune des possibilités suivantes, indiquez en le justifiant si la quantité de mémoire proposée suffit ou ne suffit pas à mémoriser une position du jeu de go : 256 bits, 800 bits, 30 octets, 100 octets.

Exercice 3 : Langage machine (4 points)

Dans cet exercice, on se place dans le cadre du microprocesseur MP0 et du langage machine LM0 décrit en cours. On rappelle que MP0 est un microprocesseur possédant deux registres d'adresses 8 bits A0 et A1 et deux registres de données 8 bits D0 et D1. LM0 possède notamment les instructions de transfert d'octet MOVE, d'addition ADD, de soustraction SUB. Ces instructions ont deux opérandes. Le premier est l'opérande source et le deuxième est l'opérande destination, qui sera modifié grâce à l'opérande source. LM0 a 3 types d'adressage: immédiat (on préfixe l'opérande avec #), direct (l'opérande est laissé tel quel) et indirect (l'opérande, qui est toujours un registre d'adresse, est mis entre parenthèses).

3.1 On considère que la mémoire contient la valeur 5 à l'adresse 50. On exécute la série d'instructions suivante stockée à partir de l'adresse 100 :

```
100 : MOVE 50, D0
102 : CMP #2, D0
104 : JGT 110
106 : SUB #2, D0
108 : JMP 102
110 : ADD D0, 50
```

Quelle valeur sera à l'adresse 50 après l'exécution de ces instructions? Expliquer pourquoi. Même question si à l'adresse 50, il y avait eu la valeur 250 à la place de la valeur 5. On rappelle que CMP est l'instruction de comparaison. L'instruction JGT est le saut conditionnel qui effectue un saut à l'adresse indiquée par son opérande si la valeur désignée par le premier opérande de l'instruction CMP (qui précède l'instruction JGT) est strictement supérieure à la valeur désignée par le deuxième opérande (de la même instruction CMP).

3.2 On exécute la série d'instructions suivante stockée à partir de l'adresse 200 :

```
200 : MOVE #100, D0
202 : PSH D0
204 : SUB #1, D0
206 : CMP #10, D0
208 : JNE 202
210 : POP D1
212 : ADD D1, D0
```

Quel est l'effet de l'exécution de ces instructions? Expliquer pourquoi. On rappelle que l'instruction PSH permet d'empiler son opérande sur la pile et que l'instruction POP permet de retirer le sommet de la pile. La pile est initialisée à l'adresse 255 au démarrage de la machine. JNE est l'instruction conditionnel de saut si les deux valeurs comparées (par l'instruction CMP) sont différentes.

Exercice 4 : Utilisation d'Unix (7 points)

4.1 On se trouve dans un répertoire ne contenant que les fichiers `bourgeois.txt` et `gentilhomme.txt`. Le fichier `bourgeois.txt` contient l'unique ligne de texte `Belle Marquise, vos beaux yeux` et le fichier `gentilhomme.txt` contient l'unique ligne de texte `Me font mourir d'amour`. Indiquez les effets du lancement des commandes Unix suivantes à partir d'un shell. Signalez les erreurs éventuelles. Chaque question est indépendante : on fera comme si la commande précédente n'avait pas été exécutée.

- a) `$ cat bourgeois.txt`
- b) `$ cat < gentilhomme.txt`
- c) `$ cat moliere.txt`
- d) `$ cat bourgeois.txt gentilhomme.txt`
- e) `$ cat bourgeois.txt > gentilhomme.txt`
- f) `$ cat bourgeois.txt >> gentilhomme.txt`
- g) `$ cat bourgeois.txt bourgeois.txt gentilhomme.txt | head -n2`
- h) `$ cat bourgeois.txt | cat`
- i) `$ cat | cat > gentilhomme.txt`
- j) `$ cat`
- k) `$ echo bourgeois.txt >> bourgeois.txt`
- l) `$ ls abc.txt > abc.txt`

4.2 On se trouve dans un répertoire `Rep` contenant initialement uniquement deux répertoires `Rep1` et `Rep2`. Le répertoire `Rep1` contient uniquement le fichier `f1`. Le répertoire `Rep2` contient uniquement le fichier `f2`. Donner les commandes qui permettent successivement d'effectuer les opérations suivantes :

- se déplacer dans le répertoire `Rep1` et changer le nom du fichier `f1` en `f`,
- en restant dans `Rep1`, changer le nom du fichier `f2` se trouvant dans `Rep2` en `f`,
- en restant dans `Rep1`, échanger le contenu des fichiers (tous les deux nommés `f`) de `Rep1` et `Rep2` (on peut utiliser plusieurs commandes),
- en restant dans `Rep1` et en une seule commande, créer dans le répertoire `Rep` le fichier `toutf` qui est la concaténation des deux fichiers nommés `f` des répertoires `Rep1` et `Rep2`,
- se déplacer dans `Rep` puis modifier le fichier `f` de `Rep1` de telle manière qu'il contienne deux fois de suite ce qu'il contenait avant (on peut utiliser plusieurs commandes).

4.3 Indiquez les commandes permettant d'effectuer les opérations suivantes, sachant que le fichier `exemple.txt` et le répertoire `Exemple` se trouve dans le répertoire courant :

- a) afficher la liste de tous les fichiers du répertoire `Exemple` dont le nom contient la suite de caractères `'abc'` (sans se déplacer dans le répertoire `Exemple`).
- b) afficher les trois premiers fichiers ou répertoires contenus dans le répertoire courant.
- c) rajouter à la fin du fichier `exemple.txt` la première ligne de ce même fichier.