Informatique 3 - DS3

Note :
Calculatrice et documents non autorisés.

Exercice 1 (*AVL*) *(7 pts)*

1. *3.5 pts* - Les éléments-ci dessous sont ajoutés/supprimés dans l'ordre dans un AVL. Dessiner l'arbre à chaque étape et indiquer les équilibres des noeuds. Si une rotation est nécessaire indiquer le type de rotation à effectuer et dessiner l'arbre avant et après cette rotation.

   Etapes de la construction de l'arbre AVL :
   a) Ajouter 15 b) Ajouter 10
   c) Ajouter 25 d) Ajouter 28
   e) Supprimer 10 f) Ajouter 26
   g) Ajouter 30 h) Ajouter 27
   i) Supprimer 15
2. *0.5 pt* - Definir la structure `Arbre` permettant de construire un AVL.
3. *3 pts* - Écrire la fonction `Double rotation gauche` ainsi que la/les fonctions de rotation qu'elle utilise le cas échéant.

Exercice 2 (*Unix : question de cours*) *(3.5 pts)*

1. *0.5 pt* - Quelle est la différence entre la racine « root / » du système de fichiers, et le répertoire de connexion de l'utilisateur ?
2. *0.5 pt* - Expliquer le fonctionnement du tube « | » utilisé par exemple dans la commande « cmd1 | cmd2 ».
3. *0.5 pt* - Quelle est la différence entre un chemin absolu et un chemin relatif ?
4. *0.5 pt* - Quels sont les différents types de fichiers dans le système de fichiers de Linux ?
5. *0.75 pt* - Expliquer comment fonctionnent les droits d'accès pour un fichier (quels sont-ils, à quoi correspondent-ils et comment les modifier).
6. *0.75 pt* - Expliquer à quoi sert la chaîne suivante « 2>&1 » dans une commande.

Exercice 3 (*Commandes Unix*) *(4.5 pts)*
Soit le fichier etudiants.txt suivant, placé dans votre répertoire de connexion.

```
12345 Marlene Culberston 4 12
22546 Donald  Pierce     3 14
15468 Barbara Culberston 3  5
18792 Diane   Freeman    2 11
20154 Steve   Tate       1 12
11458 David   Mccall     2  8
```

NB : La 1ère colonne désigne l'identifiant de l'étudiant, la 2ème son prénom, la 3ème son nom, la 4ème son année d'étude et la 5ème sa note. Pour chaque question suivante Écrire UNE ligne de commande permettant de :

1. Trier ce fichier par ordre décroissant sur l'identifiant de l'étudiant.
2. Trier ce fichier par ordre croissant sur l'année d'étude et le nom et affichez les champs nom, année d'étude et note.
3. Chercher (en ignorant la case) les étudiants de nom « Culberston » et affichez leurs détails en majuscule.

Exercice 4 (*Script shell*) *(5 pts)*

1. *(2.5 pts)* - Écrire un script `moyenne.sh` qui prend en paramètre un ou plusieurs (sans limite) nombre entre 0 et 20 et qui va
   (a) Vérifier la validité des arguments
   (b) calculer la moyenne des arguements
   (c) Afficher " redoubement" si la moyenne est en dessous de 7, "rattrapage" si elle est entre 7 et 10, "passage" si elle est au dessus de 10 et "félicitaion" si la moyenne vaut plus de 16.

2. *(2.5 pts)* - Écrire un script `cptrm.bash` qui prend en arguments deux noms de fichiers f1 et f2. Par exemple : `./cptrm.bash f1 f2`. Ce script doit réaliser les actions suivantes :
   (a) Vérifier que le nombre d'arguments passés est correct (c'est-à-dire 2) ; sinon, il affiche un message illustrant la syntaxe d'appel du script (par exemple : "Erreur ! syntaxe d'appel ./cptrm.bash f1 f2").
   (b) Chercher (en ignorant la casse) dans le répertoire de connexion et dans ses sous-répertoires, le fichier entré comme 1er argument et afficher son nombre de lignes.
   (c) Chercher dans le répertoire de connexion et dans ses sous-répertoires, le fichier entré comme 2ème argument et le supprimer.

DOCUMENTATION

---

cut - remove sections from each line of files

```
SYNOPSIS
       cut OPTION... [FILE]...
DESCRIPTION         top
       Print selected parts of lines from each FILE to standard output. With no FILE, or when
       FILE is -, read standard input. Mandatory arguments to long options are mandatory for
       short options too.
       -b, --bytes=LIST
              select only these bytes
       -c, --characters=LIST
              select only these characters
       -d, --delimiter=DELIM
              use DELIM instead of TAB for field delimiter
       -f, --fields=LIST
              select only these fields;  also print any line that contains no delimiter
              character, unless the -s option is specified
```

find - search for files in a directory hierarchy

```
SYNOPSIS
       find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...]
       [expression]
       Supported tests:
       -iname pattern
              Like -name, but the match is case insensitive.  For example, the patterns `fo*'
              and `F??' match the file names `Foo', `FOO', `foo', `fOo', etc.  The pattern
              `*foo*` will also match a file called '.foobar'.
       -inum n
              File has inode number smaller than, greater than or exactly n.  It is normally
              easier to use the -samefile test instead.
       -name pattern
              Base of file name (the path with the leading directories removed) matches shell
              pattern pattern.
       -user uname
              File is owned by user uname (numeric user ID allowed).
   ACTIONS
       -delete
              Delete files or directories; true if removal succeeded. If the removal failed,
              an error message is issued and find's exit status will be nonzero (when it
              eventually exits).
  -exec command ;
              Execute command; true if 0 status is returned.  All following arguments to find
              are taken to be arguments to the command until an argument consisting of `;' is
              encountered.  The string `{}' is replaced by the current file name being
              processed everywhere it occurs in the arguments to the command, not just in
              arguments where it is alone, as in some versions of find.
```

grep - print lines that match patterns

```
SYNOPSIS
       grep [OPTION...] PATTERNS [FILE...]
       grep [OPTION...] -e PATTERNS ... [FILE...]
       grep [OPTION...] -f PATTERN_FILE ... [FILE...]
DESCRIPTION
       grep  searches  for  PATTERNS  in  each  FILE.  PATTERNS is one or patterns separated by
       newline characters,  and  grep  prints  each line that matches a pattern.
OPTIONS
   Matching Control
       -f FILE, --file=FILE
              Obtain patterns from FILE, one per line.  If this option is used  multiple  times
              or is combined with the -e (--regexp) option, search for all  patterns  given.
              The  empty file contains zero patterns, and therefore matches nothing.
       -i, --ignore-case
              Ignore  case  distinctions,  characters that differ only in case match each other
       -v, --invert-match
              Invert the sense of matching, to select non-matching lines.
       -w, --word-regexp
              Select only those lines containing matches that form  whole words.  The test is
              that the matching substring must either be at the beginning of the line, or
              preceded by a  non-word constituent character.  Similarly, it must be either at
              the end of the line  or  followed  by  a  non-word  constituent character.
              Word-constituent   characters   are  letters,  digits, and the underscore.
              This option has no  effect  if -x is also specified.
   General Output Control
       -c, --count
              Suppress normal output; instead print a count  of  matching lines  for  each
              input  file.  With the -v, --invert-match option (see below), count
              non-matching lines.
       -m NUM, --max-count=NUM
              Stop reading a file after NUM matching lines.
   Output Line Prefix Control
       -n, --line-number
              Prefix each line of output with  the  1-based  line  number
              within its input file.
```

ls - list directory contents

```
SYNOPSIS
       ls [OPTION]... [FILE]...
DESCRIPTION
       List information about the FILEs (the current directory by default).  Sort entries
       alphabetically if none of -cftuvSUX nor --sort is specified. Mandatory arguments to
       long options are mandatory for short options too.
       -a, --all
              do not ignore entries starting with .
       -c     with -lt: sort by, and show, ctime (time of last modification of file status
              information); with -l: show ctime and sort by name; otherwise: sort by ctime,
              newest first
       -C     list entries by columns
       -d, --directory
              list directories themselves, not their contents
       -f     list all entries in directory order
       -F, --classify[=WHEN]
              append indicator (one of */=>@|) to entries WHEN
       --file-type
              likewise, except do not append '*'
       -H, --dereference-command-line
              follow symbolic links listed on the command line
       -i, --inode
              print the index number of each file
       -l     use a long listing format
       -n, --numeric-uid-gid
```

```
              like -l, but list numeric user and group IDs
       -N, --literal
              print entry names without quoting
       -o     like -l, but do not list group information
       -p, --indicator-style=slash
              append / indicator to directories
       -r, --reverse
              reverse order while sorting
       -s, --size
              print the allocated size of each file, in blocks
       -S     sort by file size, largest first
       -v     natural sort of (version) numbers within text
       -X     sort alphabetically by entry extension
```

sort - sort lines of text files

```
DESCRIPTION
     -d, --dictionary-order
              consider only blanks and alphanumeric characters
       -f, --ignore-case
              fold lower case to upper case characters
       -g, --general-numeric-sort
              compare according to general numerical value
        -n, --numeric-sort
              compare according to string numerical value
       -R, --random-sort
              shuffle, but group identical keys.  See shuf(1)
       -r, --reverse
              reverse the result of comparisons
       --sort=WORD
              sort according to WORD: general-numeric -g, human-numeric
              -h, month -M, numeric -n, random -R, version -V
```

tr - translate or delete characters

```
SYNOPSIS
       tr [OPTION]... STRING1 [STRING2]
DESCRIPTION
       Translate, squeeze, and/or delete characters from standard input,
       writing to standard output.  STRING1 and STRING2 specify arrays
       of characters ARRAY1 and ARRAY2 that control the action.
       -d, --delete
              delete characters in ARRAY1, do not translate
       -s, --squeeze-repeats
              replace each sequence of a repeated character that is
              listed in the last specified ARRAY, with a single
              occurrence of that character
        [:digit:]
              all digits
       [:lower:]
              all lower case letters
       [:upper:]
              all upper case letters
```