Licence d'informatique
Module de C/Unix

Compression et décompression de listes de mots triés

Examen – Partie Unix

Philippe MARQUET

Janvier 1999

Les documents de cours et TD sont autorisés.
On rendra deux copies séparées pour la partie Unix et pour la partie langage C. La partie Unix est à rendre sur une *copie de couleur.*

Important Plusieurs commandes Bourne-shell sont à développer. On veillera particulièrement à détailler pour chacune des commandes un algorithme avant de donner le code Bourne-shell lui-même. La notation sera répartie entre ces deux composantes de vos réponses.

Nous allons développer une commande `cmprs` qui compresse une liste de mots triée (un dictionnaire). L'option `-d` de la commande permet de décompresser le résultat fourni par `cmprs`. Une première version de la syntaxe est la suivante :

`cmprs` [-d] [*`inputfile`* [*`outputfile`*]]

Par défaut, la commande lit sur son entrée standard et produit son résultat sur la sortie standard.

La compression est réalisée par l'élimination des préfixes communs des mots et par leur remplacement par un caractère unique qui code le nombre de caractères communs avec le mot précédent.

Soit le dictionnaire `/tmp/dico`; le résultat de la commande `cmprs` est le suivant :

```
% cat /tmp/dico
computer
computerize
computerized
computerizes
computerizing
computers
inform
informal
informality
informally
informant
informants
information
informational
informative
informatively
informed
informer
informers
informing
informs
```

```
% cmprs /tmp/dico
0computer
8ize
Bd
Bs
Aing
8s
0inform
6al
8ity
8ly
7nt
9s
7tion
Bal
9ve
Bly
6ed
7r
8s
6ing
6s
```

Les caractères de codage sont les suivants :

0-9 représentent les entiers de 0 à 9 ;

A-Z représentent les entiers de 10 à 35 ;

a-z représentent les entiers de 36 à 61.

Traitement des préfixes

Deux fonctions Bourne-shell `char-to-int` et `int-to-char` permettent de passer des préfixes sous forme d'un caractère à leur valeur entière et inversement. Ces fonctions acceptent un argument et affiche leur résultat sur la sortie standard. Elles retournent normalement la valeur 0 et la valeur 1 en cas d'erreur.

Exemples :

```
% char-to-int 9
9
% char-to-int X
33
% int-to-char 37
b
```

```
% int-to-char 99

% echo $status
1
```

Question 1 (`char-to-int`)
Donner le code Bourne-shell de la commande `char-to-int`.

Question 2 (`int-to-char`)
Donner le code Bourne-shell de la commande `int-to-char`.

Indication Les exercices 1 et 2 peuvent être traitées en utilisant exclusivement les structures de contrôle du Shell et les commandes `tr`, `expr`, et `cut`.

Compression

Question 3 (`cmd-cmprs`)
Donner le code Bourne-shell de la fonction `cmd-cmprs` qui produit sur la sortie standard la compression de ce qui est lu sur l'entrée standard.

Décompression

La fonction Bourne-shell `cmd-uncmprs` produit sur sa sortie standard le résultat de la décompression de ce qui est lu sur son entrée standard.

On ne demande pas d'écrire le code de cette fonction.

La commande `cmprs`

On complète la syntaxe de la commande `cmprs` donnée en début de sujet. La nouvelle syntaxe est la suivante :

`cmprs` [`-d`] [`-z`] [*`inputfile`* [*`outputfile`*]]

Il est précisé que les options de la commande peuvent être combinées. Ainsi `-dz` ou `-zd` sont équivalents à `-d` suivi de `-z`.

Dans la syntaxe de la commande, une valeur `-` pour les fichiers *`inputfile`* ou *`outputfile`* indique de considérer l'entrée/la sortie standard. Ainsi les trois commandes suivantes sont équivalentes :

```
% cmprs -d - /tmp/toto
% cmprs -d - > /tmp/toto
% cmprs -d > /tmp/toto
```

Il est précisé que le fichier d'entrée et le fichier de sortie ne peuvent être égaux.

L'option `-z` indique que les fichiers d'entrée et de sortie sont compressés par la commande `gzip`. Ainsi les deux commandes suivantes sont équivalentes :

```
% zcat /tmp/toto.cpr.gz | cmprs -d | gzip - > /tmp/toto.gz
% cmprs -dz /tmp/toto.cpr.gz /tmp/toto.gz
```

Question 4 (`cmprs`)
Donner le code Bourne-shell de la commande `cmprs`.