

Nom : Prénom :

Contrôle du 9 février 2009

Créer, sous votre répertoire d'accueil, un sous-répertoire `controle-IP1` dans lequel vous copierez tous les fichiers fournis et en faire votre répertoire de travail. Pour toutes les questions UNIX posées, on indiquera avec précision sur l'énoncé **les commandes unix utilisées** pour obtenir le résultat demandé. Pour éviter toute ambiguïté, on exagérera nettement la longueur des espaces entre les mots.

Partie unix

Le fichier `villes.utf` (dont un extrait est reproduit ci-dessous) contient des informations sur les villes les plus peuplées dans le monde.

Abidjan	Côte_Ivoire	Afrique	3300000	298
Ahmadabad	Inde	Asie	4525000	202
Alexandrie	Egypte	Afrique	4000000	293
Alger	Algerie	Afrique	3200000	453
.
Xian	Chine	Asie	3000000	531

On dispose aussi d'un fichier `entete.txt` constitué des trois lignes suivantes indiquant la signification des colonnes :

=====				
Ville	Pays	Continent	Popul.	Surf.
=====				

- Afficher à l'écran le nombre de villes du fichier `villes.utf`
Solution
`wc -l villes.utf`
- Constituer un fichier unique `villes1.utf` sous la forme suivante à partir des fichiers `entete.txt` et `villes.utf` :

=====				
Ville	Pays	Continent	Popul.	Surf.
=====				
Abidjan	Côte_Ivoire	Afrique	3300000	298
Ahmadabad	Inde	Asie	4525000	202
Alexandrie	Egypte	Afrique	4000000	293
Alger	Algerie	Afrique	3200000	453
.
.
Xian	Chine	Asie	3000000	531

Solution

```
cat entete.txt villes.utf > villes1.utf
```

- Afficher les informations des villes dont le nom commence par T.

Solution

```
grep '^T' villes.utf
```

- Compter le nombre de villes africaines dans `villes.utf`

Solution

```
grep -c ' Afrique ' villes.utf
grep ' Afrique ' villes.utf | wc -l
awk '$3=="Afrique"{n++}END{print n}' villes.utf
```

- Afficher les informations des villes par ordre de population décroissante

Solution

```
sort -k4,4nr villes.utf
```

- Afficher les informations des 10 villes les plus peuplées par ordre de population décroissante

Solution

```
sort -k4,4nr villes.utf | head -n 10
```

7. Afficher les informations des 5 villes européennes les plus peuplées

Solution

```
grep ' Europe ' villes.utf | sort -k4,4nr | head -n 5
grep ' Europe ' villes.utf | sort -k4,4n | tail -n 5
```

8. Afficher les informations de toutes les villes en ajoutant leur densité de population en fin de ligne

Solution

```
awk '{print $0, $4/$5}' villes.utf
```

9. Afficher le nom et la population de la ville la plus peuplée

Solution

```
sort -k4,4nr villes.utf | head -n 1 | awk '{print $1, $4}'
```

10. Afficher la moyenne de la population des villes de plus de 15 millions d'habitants.

Solution

```
awk ' $4>15000000{p+=$4; n++}END{print n/p} ' villes.utf
```

Partie fortran

Le programme `pgm-utf.f90` permet de créer à partir du tableau de rang 2 `pixels1(:, :)` une image en niveaux de gris `rampe+br.pgm` présentant selon l'horizontale un dégradé du noir au blanc légèrement bruité.

Il comporte aussi des instructions en commentaire précédées de doubles points d'interrogation qu'il s'agit de décommenter au fur et à mesure que l'on complète les sous-programmes de traitement pour effectuer les opérations suivantes. A la fin de chaque étape, copier le programme opérationnel avant de commencer les modifications pour la question suivante de façon à conserver les versions déjà testées.

1. Expliquer et justifier l'ordre que doivent respecter, dans le programme principal, les trois instructions :

```
read(*,*) largeur, hauteur
allocate(pixels1(largeur, hauteur), stat=ok)
call gradient(pixels1, valmax)
```
2. Compiler ce programme avec `g2003p`, l'exécuter pour produire une image d'environ 1000 par 400 pixels. Afficher l'image `rampe+br.pgm`.
3. Compléter le sous-programme `inverse_video` pour calculer les pixels `pixels2` permettant de produire une image en vidéo inverse `rampe-inv.pgm`.
4. Reprendre ce traitement dans `inverse_moitie` en le restreignant à la moitié supérieure de l'image pour créer l'image `rampe-inv2.pgm`.
5. Compléter le sous-programme `grille` pour tracer, sur l'image initiale, une grille de 1 pixel de large avec un pas de 50 pixels en largeur et 20 pixels en hauteur. L'appel de `grille` est suivi de l'écriture de l'image `rampe+gr.pgm`.

Envoyer les fichiers sources correspondant aux trois dernières questions en fichier attaché via un seul message à l'adresse `jacques.lefrere@upmc.fr` avec pour sujet TGAE.