

ARCSYS1 : Systèmes et Réseaux

Examen final du 14 décembre 2016 (2h)

Tous documents interdits à l'exception d'une page A4 recto, manuscrite de votre main.

La correction tiendra compte de la qualité de l'argumentaire et de la présentation.

Un code illisible et/ou incompréhensible est un code faux. Le barème indiqué est approximatif.

★ Exercice 1: Chaînes de caractères et fichiers (4 pts).

On souhaite écrire un programme affichant le décompte des voyelles présentes dans une chaîne de caractères donnée.

▷ **Question 1:** Écrivez un programme complet prenant une chaîne de caractères à analyser en argument de la ligne de commande, de la manière suivante.

```
1 $ ./compte-voyelles "Bonjour. Bon courage pour ce partiel."  
2 Nombre de a: 2  
3 Nombre de e: 3  
4 Nombre de i: 1  
5 Nombre de o: 5  
6 Nombre de u: 3  
7 Nombre de y: 0
```

▷ **Question 2:** Modifiez votre programme afin qu'il lise le texte à analyser depuis un fichier dont le nom est passé en argument de la ligne de commandes.

★ Exercice 2: Programmation structurée en C (4 pts)

Un loueur de voitures décide de concevoir son propre progiciel de gestion intégré (ERP) pour gérer son parc. Chaque voiture est définie par sa catégorie (Citadine, Berline, Familiale, Utilitaire), sa plaque d'immatriculation et son prix par jour. Chaque voiture peut également être actuellement disponible, ou bien déjà louée.

▷ **Question 1:** Écrivez une structure de données `voiture_t` contenant toutes les informations relatives à une voiture donnée. Vous suivrez l'approche plutôt orientée objet vue en cours. Écrivez une fonction `voiture_new()`, prenant en arguments les informations initiales, et retournant une référence à un «objet», c'est-à-dire un pointeur sur une nouvelle structure `voiture_t`.

▷ **Question 2:** Écrivez une fonction `voiture_est_libre()` retournant si oui ou non la voiture en paramètre est libre. Écrivez `voiture_louee()` qui s'assure que la voiture en paramètre n'est plus libre.

▷ **Question 3:** Écrivez la déclaration d'une nouvelle structure nommée `parc_t`, permettant de stocker les informations relatives à toutes les voitures du loueur. Cette structure contiendra un tableau de toutes les voitures du loueur, ainsi qu'une variable stockant la taille de ce tableau.

▷ **Question 4:** Écrivez la fonction `parc_new()`, qui crée un parc locatif vide, ainsi que la fonction `parc_free()` qui libère toute la mémoire réservée par un parc locatif donné.

★ Exercice 3: Lecture de code (3pts).

Indiquez ce qu'affiche l'exécution du programme C suivant, qui compile et s'exécute sans erreur.

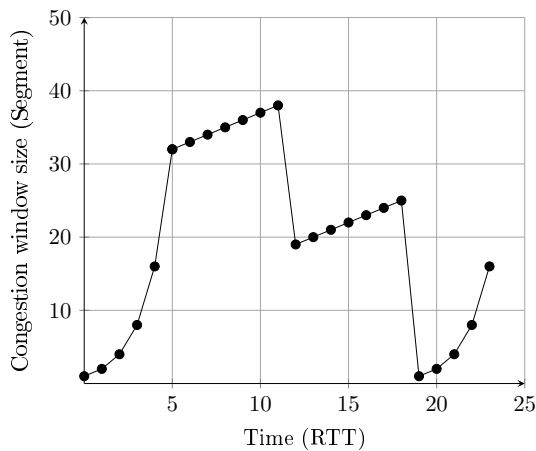
```
Programme 3.1 (début)  
1 #include <stdlib.h>  
2 #include <stdio.h>  
3 void f1 ( int a, int *b ) {  
4     a = *b;  
5 }  
6 void f2 ( int *b, int c ) {  
7     *b = c;  
8 }  
9 void f3 ( int *a, int c ) {  
10    f4 ( &a, c );  
11 }  
12 void f4 ( int **b, int a ) {  
13    **b = a;  
14 }
```

```
Programme 3.1 (fin)  
15 int main () {  
16     int x = 5, y = 7, z = 9;  
17     f1 ( x, &y );  
18     printf ( "x = %d, y = %d, z = %d\n", x, y, z );  
19     f2 ( &x, y );  
20     printf ( "x = %d, y = %d, z = %d\n", x, y, z );  
21     f3 ( &y, z );  
22     printf ( "x = %d, y = %d, z = %d\n", x, y, z );  
23     return 0;  
24 }
```

★ **Exercice 4: Questions de cours (3pts).**

- ▷ **Question 1:** L'utilisation de sockets TCP se fait typiquement à l'aide des fonctions `accept`, `bind`, `close`, `connect`, `listen`, `read`, `socket`, `write` (ordre alphabétique). Retrouvez l'ordre typique d'appel de ces fonctions dans un serveur TCP, puis dans un client TCP.
- ▷ **Question 2:** Dans un article récent de PCWorld.fr à propos des «DNS menteurs», on peut lire : «Il s'agit d'[...]éviter à un internaute de tomber sur une erreur 404 affichée par son navigateur Internet si le domaine auquel il essaye de se connecter n'existe pas». Expliquez pourquoi l'auteur de la phrase n'a pas l'air de comprendre correctement l'interaction entre DNS et HTTP (d'après S. Thibault).
- ▷ **Question 3:** Le protocole RTSP est un protocole réseau de la couche application spécialisé dans le streaming de vidéo, optimisant la transmission temps réel type vidéoconférence, et la diffusion à la demande. D'après vous, pourquoi les grands hébergeurs de contenu vidéo en ligne utilisent-ils exclusivement HTTP au lieu d'un tel protocole, pourtant spécifiquement optimisé pour ce type d'usage?

★ **Exercice 5: Contrôle de congestion (3pts).**



Le schéma ci-contre représente l'évolution de la taille de la fenêtre au cours du temps. Cette taille est mesurée en nombre maximal de segments potentiellement en transit sur le réseau. Le temps est mesuré en multiples de RTT, qui est le le délai de transmission pour envoyer un paquet plus le délai pour recevoir le ACK.

- ▷ **Question 1:** Indentifiez les intervalles de temps où les algorithmes de *slow start* et de *congestion avoidance* sont en action.
- ▷ **Question 2:** Que s'est-il passé coté émetteur aux temps 5, 11 et 18?
- ▷ **Question 3:** Comment la taille de la fenêtre évoluera-t-elle après le temps 23 en l'absence de nouveau problème? Pourquoi et jusqu'à quand?

★ **Exercice 6: Routage IP (3pts).**

Le schéma ci-contre représente une petite partie d'internet composée de deux sous-réseaux notés A et B. Ces deux réseaux sont interconnectés par le routeur R1, tandis que le routeur R2 interconnecte le réseau B au reste du réseau Internet.

- ▷ **Question 1:** Quel est le masque du réseau A, et combien d'hôtes ce réseau peut-il contenir au maximum?
- ▷ **Question 2:** Sachant que l'institution utilisant cette partie du réseau dispose de la plage d'adresse 128.96.36.0/22, et que l'on souhaite doter ce réseau du masque 255.255.255.128, quelle adresse devrait avoir le réseau B?
- ▷ **Question 3:** Donnez une table de routage fonctionnelle pour le routeur 1, en précisant à chaque ligne le réseau, le masque et le next hop.

