

Exercice 11

```
#include <stdio.h>
```

Voici le programme c étendu de l'exercice 10 qui ajoute puis retire les couches du modele OSI pour lire le message

```
//prototypes des fonctions
```

```
void emetteur_couche1(char* message);
```

```
void emetteur_couche2(char* message);
```

```
void emetteur_couche3(char* message);
```

```
void emetteur_couche4(char* message);
```

```
void emetteur_couche5(char* message);
```

```
void emetteur_couche6(char* message);
```

```
void emetteur_couche7(char* message);
```

```
void recepateur_couche1(char* message);
```

```
void recepateur_couche2(char* message);
```

```
void recepateur_couche3(char* message);
```

```
void recepateur_couche4(char* message);
```

```
void recepateur_couche5(char* message);
```

```
void recepateur_couche6(char* message);
```

```
void recepateur_couche7(char* message);
```

```
// emmeteur, couche 1 appelle la couche 1 du côté recepateur
```

```
void emetteur_couche1(char* message) {
```

```
    printf("Emetteur Couche 1: %s\n", message);
```

```
    recepateur_couche1(message);
```

```
}
```

```
void emetteur_couche2(char* message) {  
    char modifiedMessage[256];  
    sprintf(modifiedMessage, "COUCHE2: %s", message);  
    printf("Emetteur Couche 2: %s\n", modifiedMessage);  
    emetteur_couche1(modifiedMessage);  
}
```

```
void emetteur_couche3(char* message) {  
    char modifiedMessage[256];  
    sprintf(modifiedMessage, "COUCHE3: %s", message);  
    printf("Emetteur Couche 3: %s\n", modifiedMessage);  
    emetteur_couche2(modifiedMessage);  
}
```

```
void emetteur_couche4(char* message) {  
    char modifiedMessage[256];  
    sprintf(modifiedMessage, "COUCHE4: %s", message);  
    printf("Emetteur Couche 4: %s\n", modifiedMessage);  
    emetteur_couche3(modifiedMessage);  
}
```

```
void emetteur_couche5(char* message) {  
    char modifiedMessage[256];  
    sprintf(modifiedMessage, "COUCHE5: %s", message);  
    printf("Emetteur Couche 5: %s\n", modifiedMessage);  
    emetteur_couche4(modifiedMessage);  
}
```

```
void emetteur_couche6(char* message) {  
    char modifiedMessage[256];  
    sprintf(modifiedMessage, "COUCHE6: %s", message);  
    printf("Emetteur Couche 6: %s\n", modifiedMessage);  
    emetteur_couche5(modifiedMessage);  
}
```

```
void emetteur_couche7(char* message) {  
    char modifiedMessage[256];  
    sprintf(modifiedMessage, "COUCHE7: %s", message);  
    printf("Emetteur Couche 7: %s\n", modifiedMessage);  
    emetteur_couche6(modifiedMessage);  
}
```

```
void recepteur_couche1(char* message) {  
    printf("Recepteur Couche 1: %s\n", message);  
    recepteur_couche2(message);  
}
```

```
void recepteur_couche2(char* message) {  
    char extractedMessage[256];  
    sscanf(message, "COUCHE2: %[^\\n]", extractedMessage);  
    printf("Recepteur Couche 2: %s\n", extractedMessage);  
    recepteur_couche3(extractedMessage);  
}
```

```
void recepneur_couche3(char* message) {  
    char extractedMessage[256];  
    sscanf(message, "COUCHE3: %[^\\n]", extractedMessage);  
    printf("Recepteur Couche 3: %s\\n", extractedMessage);  
    recepneur_couche4(extractedMessage);  
}
```

```
void recepneur_couche4(char* message) {  
    char extractedMessage[256];  
    sscanf(message, "COUCHE4: %[^\\n]", extractedMessage);  
    printf("Recepteur Couche 4: %s\\n", extractedMessage);  
    recepneur_couche5(extractedMessage);  
}
```

```
void recepneur_couche5(char* message) {  
    char extractedMessage[256];  
    sscanf(message, "COUCHE5: %[^\\n]", extractedMessage);  
    printf("Recepteur Couche 5: %s\\n", extractedMessage);  
    recepneur_couche6(extractedMessage);  
}
```

```
void recepneur_couche6(char* message) {  
    char extractedMessage[256];  
    sscanf(message, "COUCHE6: %[^\\n]", extractedMessage);  
    printf("Recepteur Couche 6: %s\\n", extractedMessage);  
    recepneur_couche7(extractedMessage);  
}
```

```

void recepateur_couche7(char* message) {
    char extractedMessage[256];
    sscanf(message, "COUCHE7: %[^\\n]", extractedMessage);
    printf("Recepteur Couche 7 (Application): %s\\n", extractedMessage);
}

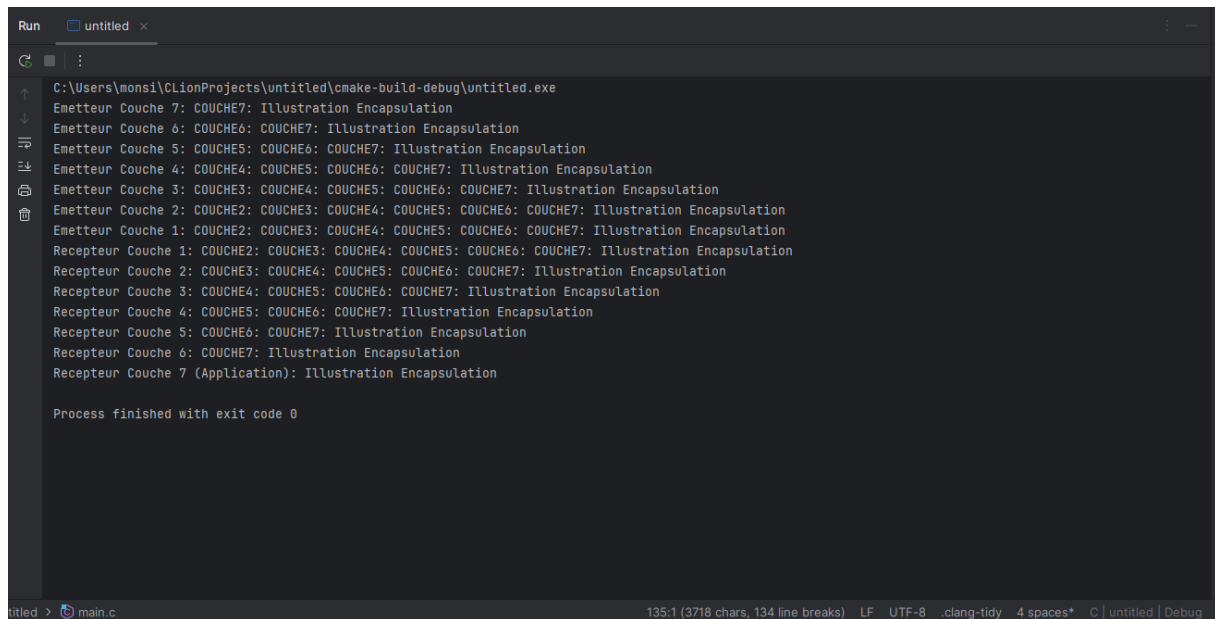
```

```

int main() {
    char message[] = "Illustration Encapsulation";
    emetteur_couche7(message);
    return 0;
}

```

Voici le resultat de l'exécution du programme c :



```

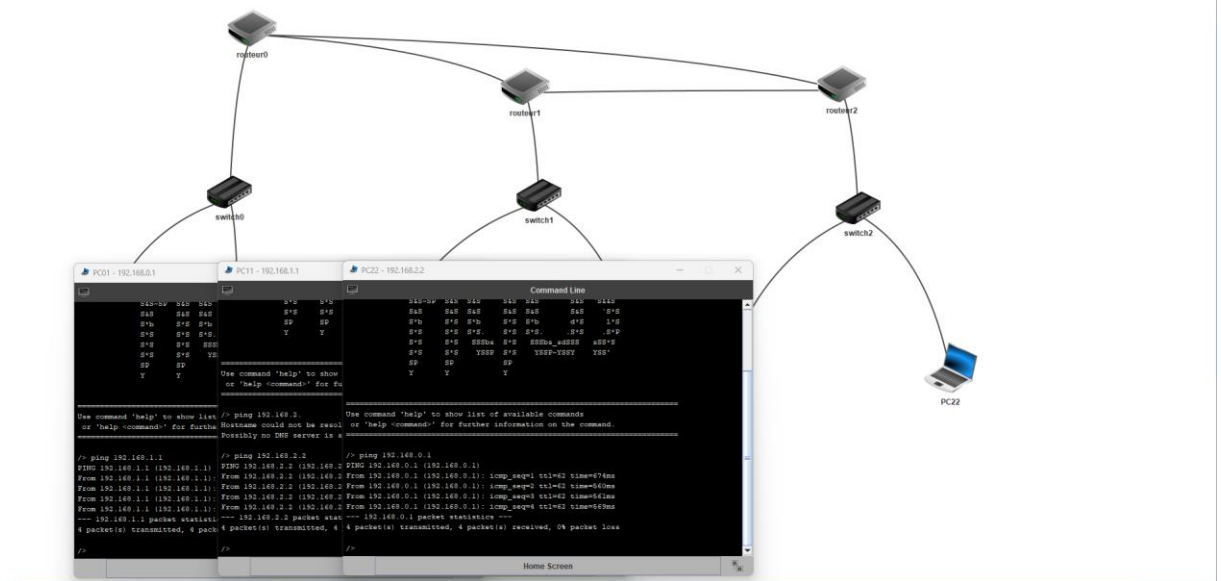
Run  untitled x
C:\Users\monsi\CLionProjects\untitled\cmake-build-debug\untitled.exe
Emetteur Couche 7: COUCHE7: Illustration Encapsulation
Emetteur Couche 6: COUCHE6: COUCHE7: Illustration Encapsulation
Emetteur Couche 5: COUCHE5: COUCHE6: COUCHE7: Illustration Encapsulation
Emetteur Couche 4: COUCHE4: COUCHE5: COUCHE6: COUCHE7: Illustration Encapsulation
Emetteur Couche 3: COUCHE3: COUCHE4: COUCHE5: COUCHE6: COUCHE7: Illustration Encapsulation
Emetteur Couche 2: COUCHE2: COUCHE3: COUCHE4: COUCHE5: COUCHE6: COUCHE7: Illustration Encapsulation
Emetteur Couche 1: COUCHE2: COUCHE3: COUCHE4: COUCHE5: COUCHE6: COUCHE7: Illustration Encapsulation
Recepteur Couche 1: COUCHE2: COUCHE3: COUCHE4: COUCHE5: COUCHE6: COUCHE7: Illustration Encapsulation
Recepteur Couche 2: COUCHE3: COUCHE4: COUCHE5: COUCHE6: COUCHE7: Illustration Encapsulation
Recepteur Couche 3: COUCHE4: COUCHE5: COUCHE6: COUCHE7: Illustration Encapsulation
Recepteur Couche 4: COUCHE5: COUCHE6: COUCHE7: Illustration Encapsulation
Recepteur Couche 5: COUCHE6: COUCHE7: Illustration Encapsulation
Recepteur Couche 6: COUCHE7: Illustration Encapsulation
Recepteur Couche 7 (Application): Illustration Encapsulation

Process finished with exit code 0

```

Exercice 12 :

a) Les pings fonctionnent sur toutes les machines :



b) Avec les tables routées manuellement les pings fonctionnent, voici les tables de routage

Routeur 0 :

Forwarding table			
Destination	Netmask	Next gateway	NIC
10.0.2.1	255.255.255.255	127.0.0.1	127.0.0.1
10.0.0.1	255.255.255.255	127.0.0.1	127.0.0.1
192.168.0.3	255.255.255.255	127.0.0.1	127.0.0.1
10.0.2.0	255.255.255.0	10.0.2.1	10.0.2.1
10.0.0.0	255.255.255.0	10.0.0.1	10.0.0.1
192.168.0.0	255.255.255.0	192.168.0.3	192.168.0.3
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1
192.168.1.1	255.255.255.255	10.0.0.4	10.0.0.1
192.168.1.2	255.255.255.255	10.0.0.4	10.0.0.1
192.168.2.1	255.255.255.255	10.0.2.2	10.0.2.1
192.168.2.2	255.255.255.255	10.0.2.2	10.0.2.1

Routeur 1 :

Forwarding table				×
Destination	Netmask	Next gateway	NIC	
10.0.1.1	255.255.255.255	127.0.0.1	127.0.0.1	
10.0.0.4	255.255.255.255	127.0.0.1	127.0.0.1	
192.168.1.3	255.255.255.255	127.0.0.1	127.0.0.1	
10.0.1.0	255.255.255.0	10.0.1.1	10.0.1.1	
10.0.0.0	255.255.255.0	10.0.0.4	10.0.0.4	
192.168.1.0	255.255.255.0	192.168.1.3	192.168.1.3	
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	
192.168.0.1	255.255.255.255	10.0.0.1	10.0.0.4	
192.168.0.2	255.255.255.255	10.0.0.1	10.0.0.4	
192.168.2.1	255.255.255.255	10.0.1.2	10.0.1.1	
192.168.2.2	255.255.255.255	10.0.1.2	10.0.1.1	

Routeur 2 :

Forwarding table				×
Destination	Netmask	Next gateway	NIC	
10.0.1.2	255.255.255.255	127.0.0.1	127.0.0.1	
10.0.2.2	255.255.255.255	127.0.0.1	127.0.0.1	
192.168.2.3	255.255.255.255	127.0.0.1	127.0.0.1	
10.0.1.0	255.255.255.0	10.0.1.2	10.0.1.2	
10.0.2.0	255.255.255.0	10.0.2.2	10.0.2.2	
192.168.2.0	255.255.255.0	192.168.2.3	192.168.2.3	
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	
192.168.0.1	255.255.255.255	10.0.2.1	10.0.2.2	
192.168.0.2	255.255.255.255	10.0.2.1	10.0.2.2	
192.168.1.1	255.255.255.255	10.0.1.1	10.0.1.2	
192.168.1.2	255.255.255.255	10.0.1.1	10.0.1.2	