

I2010 : Examen de langage C

A. Legrand – J. Vander Meulen – G. Seront

Informatique de Gestion - Bloc 2

Durée de l'examen : 2h30

Consignes importantes :

- **Compilez votre code avec les flags utilisés lors du cours :**

```
gcc -std=c11 -pedantic -Wall -Wvla -Werror
```

1. Module *video*

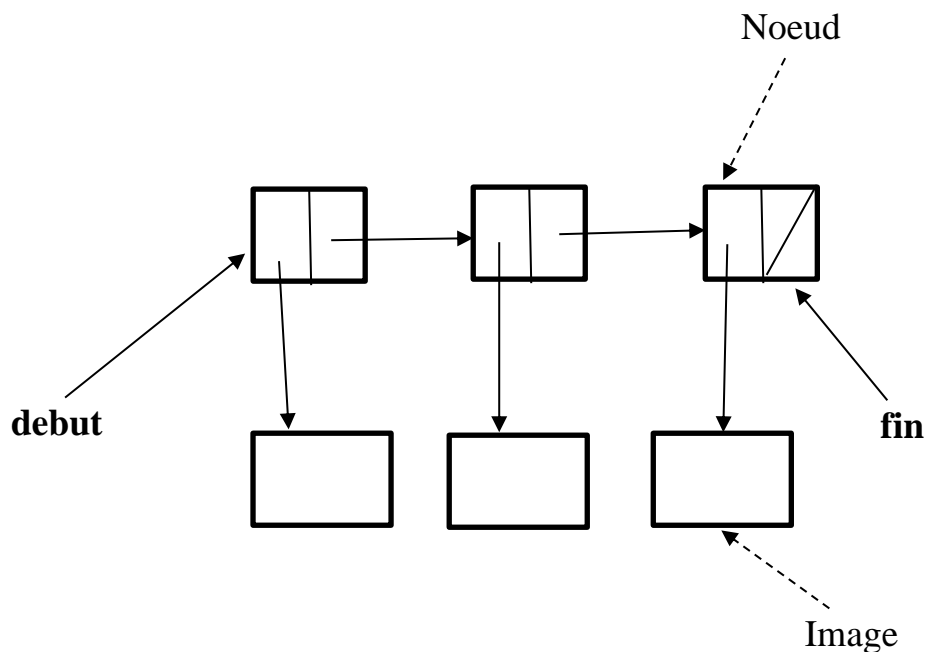
Le module **video** permet de créer des vidéos basiques, fonctionnant comme le format « GIF » animé, omniprésent sur le web. Une *Video* définit une séquence d'images et un temps de pause entre deux images. Voici sa définition de type, issue du header *video.h* :

```
typedef struct Noeud {
    Image* image;
    struct Noeud* suivant;
} Noeud;

typedef struct Video {
    Noeud* debut; // première image de la vidéo
    Noeud* fin;   // dernière image de la vidéo
    int nbImg;    // nombre d'images
    int msec;     // pause entre deux images (en millisecondes)
} Video;
```

Dans cet examen nous ne nous intéresserons pas à la partie image. La définition de la structure *Image* est fournie dans *image.h*. Elles sont représentées par une structure contenant un nom et les dimensions de l'image.

La suite d'images est implémentée à l'aide d'une liste chaînée, telle que représentée ci-dessous :



Lorsqu'une vidéo est jouée, il y a une pause entre deux images. Si une image est en position n dans la vidéo, elle sera jouée au temps $n * msec$. La première image porte le numéro zéro.

La fonction `jouerVideo()` est fournie.

Fonctionnalités à implémenter

Les spécifications et les signature des fonctions à implémenter sont fournies dans le header *video.h* sauf pour *libererVideo* pour lesquelles seules les spécifications sont fournies. Vous devez donc écrire les signatures de ces deux fonctions et faire leur appel correct dans les tests.

Video creerVideo (int temps)

Cette fonction crée un vidéo vide cad ne contenant pas d'image. Les pointeurs de début et de fin sont donc NULL. Le temps passé en paramètre est le temps de pause entre chaque image.

bool ajouterImage (Video* vid, Image* im)

Cette fonction ajoute une image à la fin d'une vidéo. Autrement dit, elle insère l'image à la fin de la liste chaînée. La fonction renvoie vrai si l'opération s'est déroulée correctement ; faux sinon.

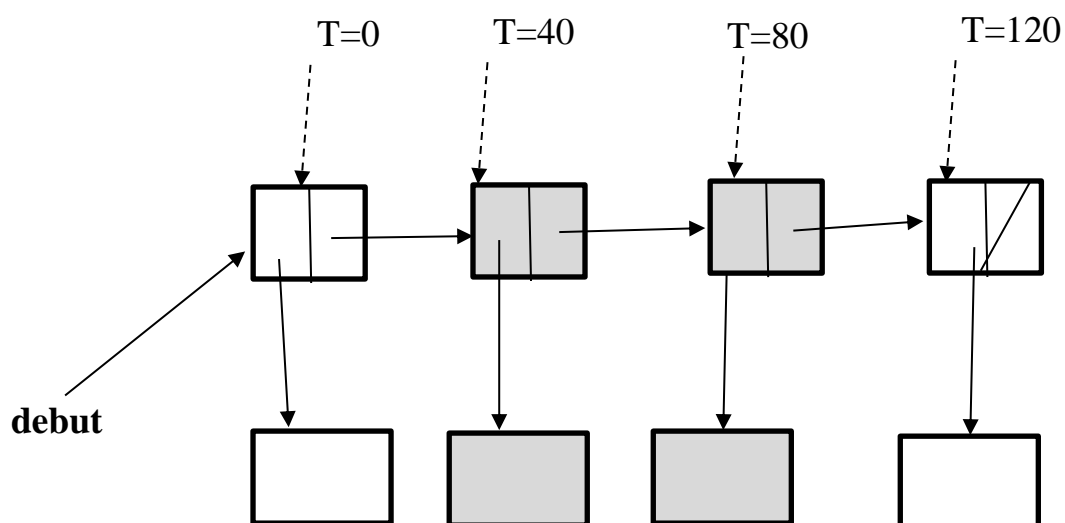
void libererVideo

Cette fonction libère la mémoire associée à une vidéo. Après traitement, la vidéo est donc vide : elle ne contient plus aucune image (elles ont toutes été libérées). La vidéo revient au même état que celui d'une vidéo nouvellement créée à l'aide de la fonction *creerVideo()*.

bool copier (Video *source, Video *clipboard, int tDebut, int tfin)

Remplit le clipboard d'une **copie** des images provenant de la source dont la position ¹ dans le temps est $\geq tDebut$ et $\leq tFin$. Si $tFin$ est supérieur à la longueur de la vidéo, on copie toutes les images de $tDebut$ jusqu'à la fin. $tDebut$ et $tFin$ ne doivent pas forcément correspondre exactement à la position d'une image.

Par exemple pour la liste ci-dessous, si $msec = 40$ et on copie $tDebut = 35$, $tFin = 100$, on copiera les deux images grisées.



¹ Notez que les positions sont ici définies par un temps.

Si aucune image n'est sélectionnée, on renvoie tout de même true (ex. tDebut = 45 et tFin = 60) en mettant nbImg de la vidéo clipboard à 0.

Le clipboard reçoit également la vitesse de la vidéo.

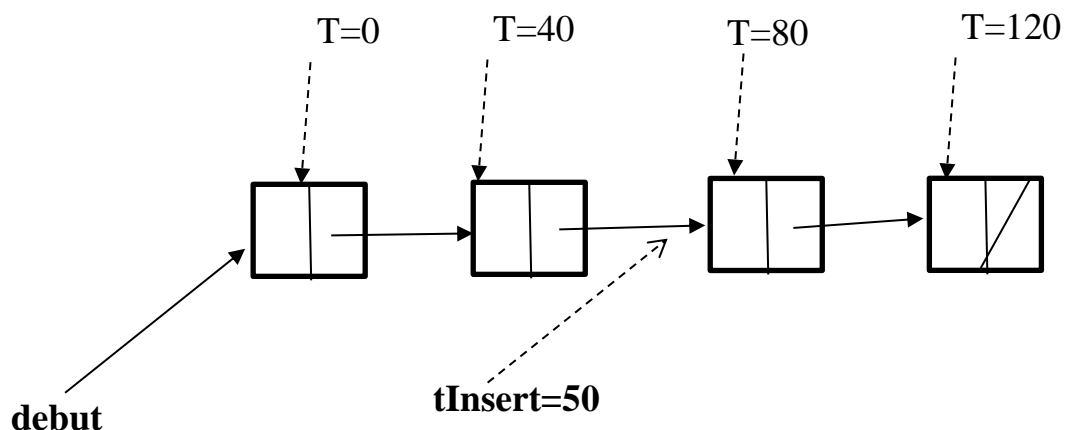
La fonction renvoie vrai si l'opération s'est déroulée correctement ; faux sinon.

Nous vous conseillons d'utiliser les fonctions creerImage et ajouterImage pour réaliser la copie des images.

bool coller

Insère les images extraites de la vidéo source dans la vidéo destination à un temps d'insertion tInsert donné. Les images sont insérées après la dernière image dont la position dans le temps est $\leq tInsert$. Si tInsert est \geq à la fin de la vidéo, on insérera à la fin.

Par exemple pour la vidéo destination ci-dessous contenant 4 images, si msec = 40 et tInsert = 50, on insère après la 2^{ème} image.



Si la vidéo source contient 3 images, la vidéo destination contiendra $4+3 = 7$ images après insertion.

On insère les images provenant de la source **sans faire de copie**. Après cette opération, la vidéo source revient au même état que celui d'une vidéo nouvellement créée à l'aide de la fonction creerVideo().

Si les msec des vidéos ne correspondent pas, la copie échouera.

La fonction renvoie vrai si l'opération s'est déroulée correctement ; faux sinon.

Programme de test & makefile

Le programme de test *testVideo.c* permet de tester les fonctionnalités du module *video*. Le programme n'est que partiellement rempli. Il vous est demandé de :

- vérifier qu'il y a bien deux arguments sur la ligne de commande :

```
./testVideo 10 300
```

Si ce n'est pas le cas, afficher un message d'erreur et terminer l'exécution ;

- récupérer les deux arguments du programme : le nombre d'images de la vidéo de test et son temps de pause en millisecondes ;
- supprimer la vidéo en fin de programme à l'aide de *libererVideo()*.

Pour finir, il vous est demandé de compléter votre fichier *makefile* afin qu'il génère l'exécutable *testVideo*.