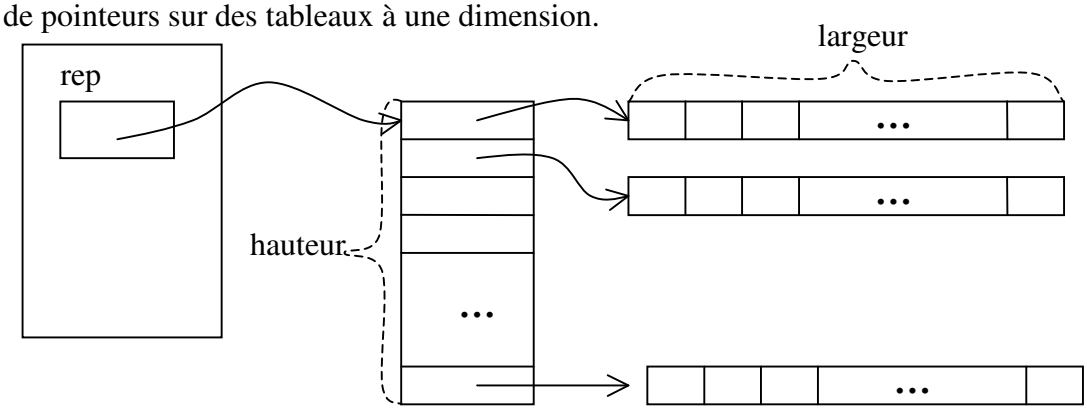
programmation
examen partiel

Partie 1. Un modèle de classe tableau à deux dimensions.

```
template <class T> class tab2{
       T**rep;
       int hauteur, largeur;
    public :
       tab2(int h, int l);
       ~tab2();
       T*& operator[](int i);
       T& operator()(int i , int j);
};
```

Nous nous proposons de représenter un tableau à deux dimensions à l'aide d'un tableau de pointeurs sur des tableaux à une dimension.



Question 1.(3 points)
Ecrire la définition du constructeur et du destructeur du modèle de classe *tab2*.

Question 2.(3 points)
Ecrire la définition des opérateurs *T*& operator[](int i);* et *T& operator()(int i , int j);* l'opérateur [] renvoie une référence à la $i^{ème}$ ligne du tableau, et l'opérateur () renvoie une référence à l'élément de ligne *i* et de colonne *j*.

Partie 2. Une classe Image.

Nous utilisons la classe *tab2* pour représenter une image : Une image est un tableau à deux dimensions de pixels, et un pixel est un entier représenté sur quatre octets, l'octet de gauche est inutilisé, les octets suivants représentent le bleu, le vert puis le rouge.

```
class Image{
     tab2<int> *image;
     int hauteur, largeur;
     void ajout( int i, int j, int & r, int &v, int & b, int & n);
   public :
      Image ( int  h, int l);
```

```
      ~Image ();
      void renverse();
      void noirEtBlanc();
      void flou();
      void sauve( char * nomFich);
};
```

Question 3.(3 points)
Ecrire la définition du constructeur et du destructeur de la classe *Image*.

Question 4.(3 points)
Ecrire la définition de la fonction *void renverse()* de la classe *Image* qui échange le haut et le bas de l'image.

Question 5.(3 points)
Ecrire la définition de la fonction *void noirEtBlanc().* Qui transforme une image en noir et blanc. Pour réaliser ceci, chaque couleur de chaque pixel est remplacé par par la moyenne des couleurs de ce pixel : si un pixel est constitué de *r*, *v*, *b* alors ses octets rouge, vert et bleu sont remplacé par *(r+v+b)/3*.

Question 6.(3 points)
Ecrire la définition de la fonction *void flou().* Qui rend une image floue : pour rendre une image floue, on remplace chaque pixel par la moyenne du pixel, et des huit pixels voisin. Cette moyenne doit se faire indépendamment sur chacune des couleurs. Pour Réaliser cette fonction, il peut être utile de définir la fonction *void ajout(int i, int j, int & r, int &v, int & b, int & n);* dont le but est d'ajouter à *r*, *v*, et *b* la composante de même couleur du pixel se trouvant *en (i,j)* s'il existe, et dans ce cas ajouter 1 à *n*. Cette fonction permet de traiter de la même façon tous les pixels de l'image, y compris ceux qui sont sur les bords et qui n'ont pas huit voisins.

Question 7.(3 points)
Ecrire la définition de la fonction *void sauve(char * nomFich);* Qui sauvegarde une image dans un fichier. Le fichier sera constitué de deux entiers, la hauteur et la largeur de l'image, puis d'une suite de couples de deux entiers *(n, c)* dans lesquels *n* est le nombre d'occurrences successives de la couleur *c*.
Exemple :

v	v	v	b
b	r	r	b
v	v	v	v

Est sauvé dans le fichier en :
3 4 3 v 2 b 2 r 1 b 4 v