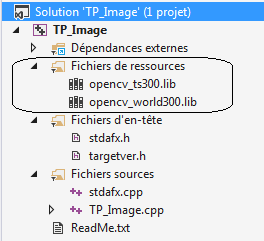
1. ***PREPARATION DE L'ESPACE DE TRAVAIL* :** (Fichiers disponiblesdans fichier rar joint)

❒ *Créer un projet type* « Console Win32 » sous Visual Studio. Le projet s’appellera **TP\_Image\_votreNom**.

***Attention*** : Lors de la création du projet, ne pas valider « Créer un répertoire pour la solution »

❒ *Copier sous le répertoire du projet* les répertoires ressources suivants:

* lib (*qui contient les bibliothèques*)
* opencv2 (*qui contient les fichiers d'en-tête*)

❒ *Ajouter dans l'explorateur de solutions* (*au niveau des « fichiers ressources »*), les librairies d'Open CV qui vont participer au projet (dans dossier /lib/ fourni, *opencv\_ts300.lib et opencv\_world300.lib*) :

❒ *Copier/Coller le fichier* source **TP\_Image\_Eval.cpp** dans le répertoire du projet et l’inclure dans votre projet. Ce sera votre fichier de travail.

❒ *Copier/Coller l'image à traiter* **« fleur.jpg »** dans le répertoire du projet.

❒ *Générer le projet* en mode « debug ».

***Attention***: Cette opération va « planter » mais va créer dans le répertoire de travail un dossier debug.

❒ *Copier/Coller la DLL* **opencv\_world300.DLL** dans le répertoire **debug**.

❒ *Lancer à nouveau l'application* pour valider l’ensemble. L’image à traiter s’affiche.

Le fichier **TP\_Image\_Eval.cpp** contient déjà une partie du code de l'application à développer, plus précisément l'appel des fonctions d'Open CV permettant de charger l'image « fleur.jpg » en mémoire et de l'afficher.

*Variables globales contenant des informations utilisables :*

* **nbRows** : nombre de lignes (*hauteur*) de l'image
* **nbCols** : nombre de colonnes (*largeur*) de l'image
* **pixelsArray[]**: tableau unidimensionnel des valeurs en NdG des pixels de l'image (seule structure utilisée pour l’affichage)
* **Histogramme[]:** Structure de données de l’histogramme ne contenant que des valeurs à 0.
* ***Toutes les autres variables que vous serez amené à utiliser devront être locales***.

***Vous coderez les questions suivantes aux emplacements réservés dans le fichier source. Ces emplacements sont repérés par des commentaires de type « //Question xx »***

1. ***TAILLE DE L’IMAGE :***
   1. Proposer un programme permettant de connaitre la hauteur et la largeur de l’image, ainsi que le nombre de pixels de l’image.
   2. Noter ces valeurs.

* nbRows = 300
* nbCols = 324
  1. Préciser la taille (en nbre d’éléments) du tableau pixelsArray[].

97 200

1. ***PROPRIETES SUR LES PIXELS :***

*Question :* Proposer un organigramme et un programme qui permet d'afficher les valeurs de NdG ***des 50 premiers pixels*** de ***la 3ère ligne de l’image.***

i = 0

i < 50

Afficher pixelsArrays[2][i]

i = i + 1

1. ***TRAVAIL SUR L’HISTOGRAMME :***

*Données à considérer :*

* *calcHisto()*: Fonction calculant et affichant les valeurs de l’histogramme.

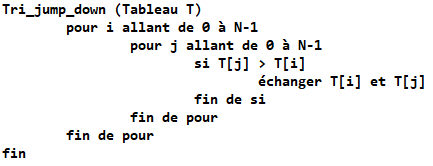
*Son prototype est déjà dans le main() et le codage de cette fonction est à la suite du main()*.

*Question :* Mettre en œuvre une fonction ***pixelMax()*** qui affiche la valeur de Niveau de Gris la plus présente dans l’image, ainsi que le nombre de pixels correspondants.

* Cette fonction ne passe pas de paramètres et ne renvoie aucune valeur.
* Avant l’appel de cette fonction, la structure *histogramme[]* devra être remplie des valeurs de l’histogramme de l’image concernée.

1. ***TRI JUMP-DOWN de l’histogramme :***

Tout comme le tri à bulles, ce tri fait remonter en premier les plus grands éléments. Toutefois, il ne travaille pas sur des éléments adjacents. Il compare chaque élément du tableau avec celui qui est à la place du plus grand, et échange lorsqu'il trouve un nouveau plus grand.



*L’algorithme lié à ce tri est le suivant :*

***T*** est le tableau à trier.

***N*** est la taille du tableau

*Données à mettre en œuvre :*

* ***triJD()****:* Fonction qui trie et affiche les valeurs triées.
* *HistoTriJD[] :* Structure de données de travail identique à *histogramme[],* ***locale*** à *triJD()*.

***Doit contenir initialement une copie de la structure histogramme[]***

*Question :* Mettre en œuvre la fonction ***triJD()***et tester la programmation.