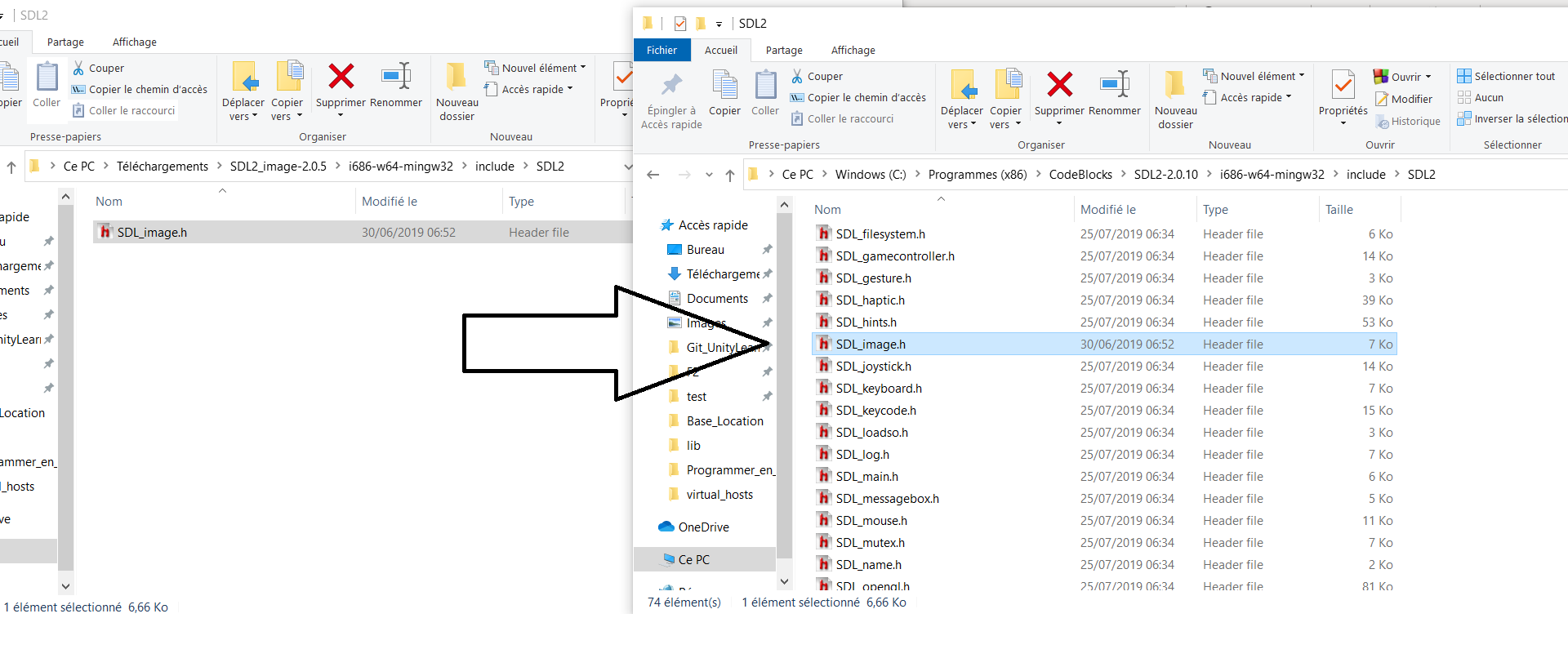
**1) Quel Api faut-il ajouter à votre projet de développement afin de pouvoir prendre en charge les images au format (JPG, PNG….), que fait cet API? 10 points**

L’API à ajouter au projet s’appelle SDL\_Image : elle charge les images en tant que surfaces et textures. Les format supportés sont les suivants : BMP, BIG, JPEG, LBM, PCX, PNG, PNM, SVG, TGA, TIFF, WEBP, XCF, XPM, XV.

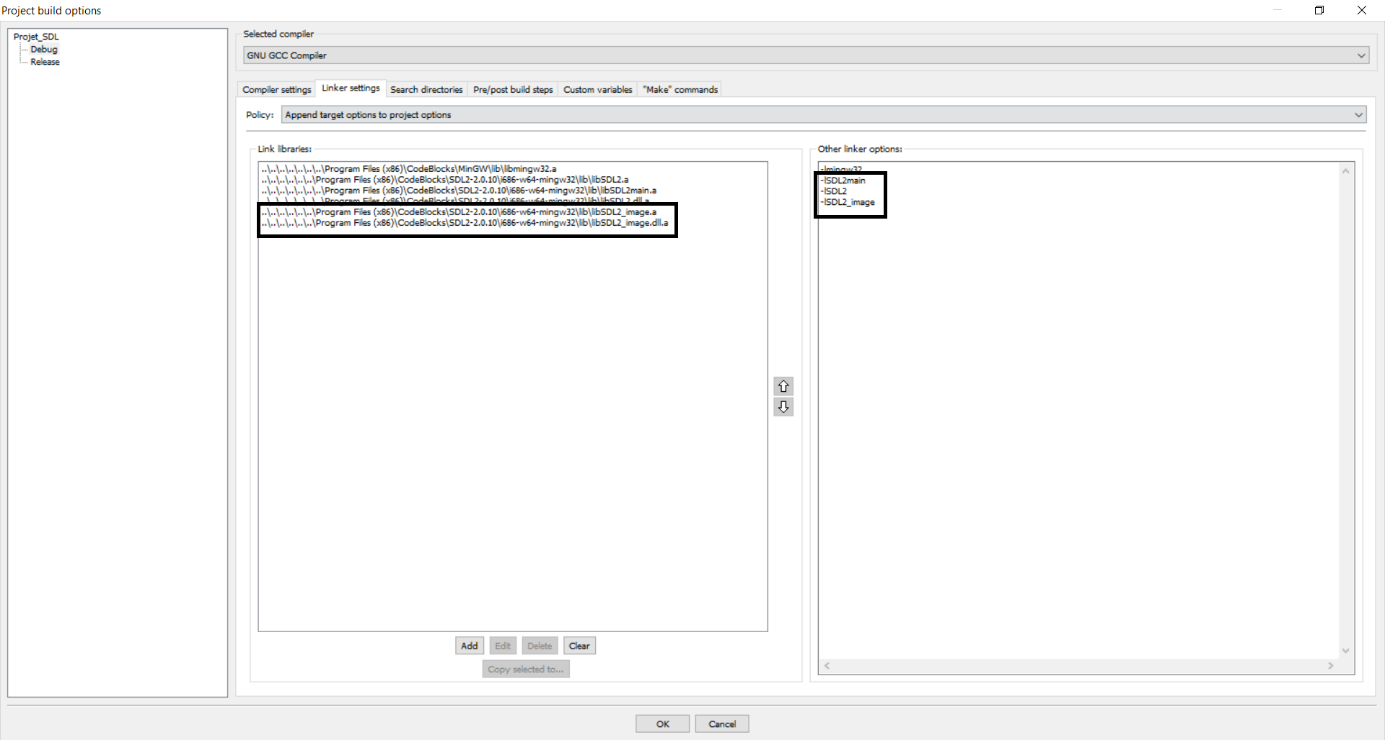
**2) Donnez les étapes de configuration pour l’utilisation de ce nouvel API. 10 points**

- Se rendre sur le site : <https://www.libsdl.org/projects/SDL_image/>  
- Télécharger « SDL2\_image-devel-2.0.5-mingw.tar.gz (MinGW 32/64-bit) » (Development Libraries)  
- Copier le contenu de chaque dossier contenu dans « i686-w64-mingw32 » dans le dossier du même nom contenu au sein du dossier SDL-2.0.10   
- Copier toutes les .dll contenues dans le dossier « bin » directement à la racine du projet



*Exemple avec le fichier SDL\_image.h*

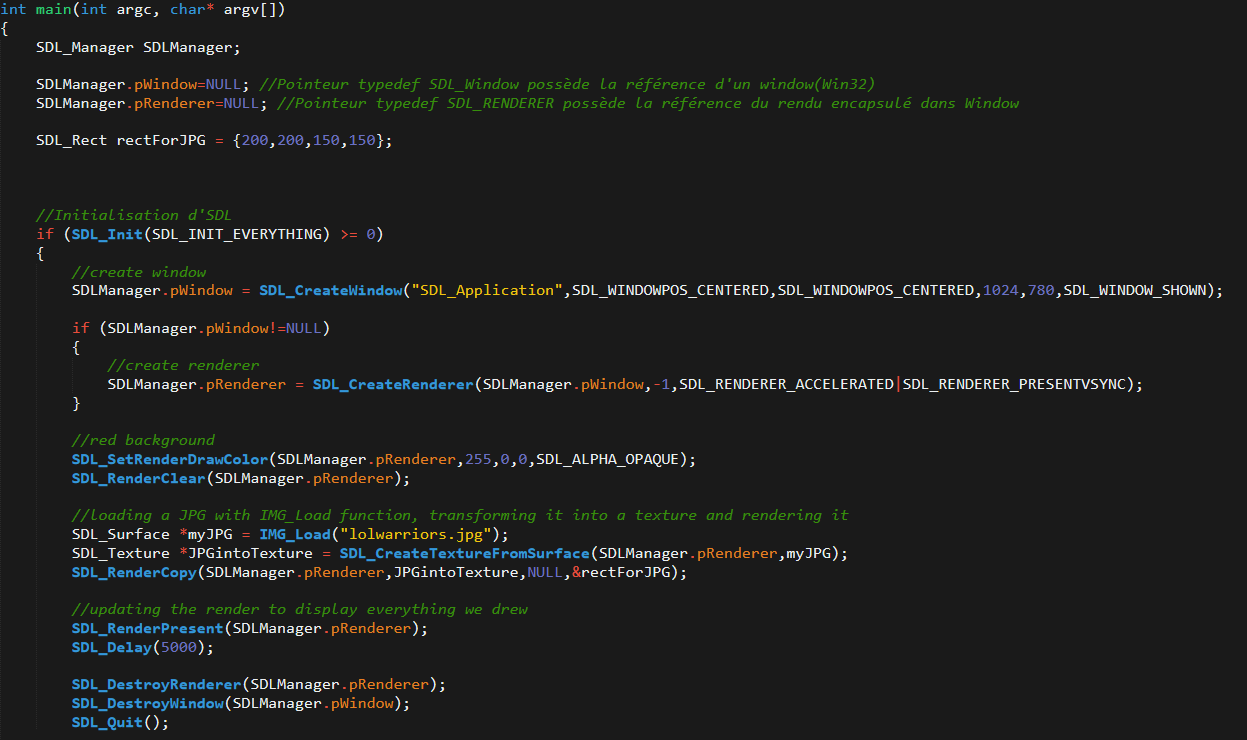
- Dans CodeBlocks, aller dans « Project => Build Options », cliquer sur « Debug » et aller dans « Linker settings »

  
Rajouter les deux blocs encadrés en noir

Et ne pas oublier « #include <SDL2/SDL\_Image.h> » dans le projet !

3) **Donnez un exemple de code en c chargeant une image au format png dans la fenêtre (image de votre choix). Commentez et expliquez ce code. 10 points**

J’ai fais pour une image JPG mais c’est le même principe.



Ca marche pareil que lors de la création d’une surface : on utilise la fonction IMG\_Load en passant en paramètre le chemin vers l’image à charger. Elle est chargée en tant que surface : on la transforme en texture et on utilise SDL\_RenderCopy pour la passer dans le render, et on l’affiche en appelant SDL\_RenderPresent().