

## Feuille de TP n°6 :

### Premier programme C++ / wxWidgets écrit 'from scratch' = sans générateur d'interface

#### Objectif pédagogique

Cette séance de TP a pour but de vous familiariser avec le framework wxWidgets (hiérarchie de classes et méthodes) permettant de créer un programme avec interface graphique.

#### TRAVAUX A RENDRE

- Réponses aux questions de la feuille de td n°5
- Sources du programme deux

À déposer **individuellement** sur l'espace Travaux avant lundi 16/05/2016, 23h59.

#### RESSOURCES A VOTRE DISPOSITION POUR REALISER LE TP

- ~ Cette feuille de TP n°6.
- ~ Ressources disponibles sur le WebCampus :
  - ~ fichiers source du programme zero.exe
  - ~ documentation wxWidgets : partielle (.pdf à utiliser en td), et complète (wx.chm à utiliser en tp)

#### DIRECTIVES PARTICULIERES A CETTE SEANCE

Dans votre répertoire de travail habituel, **et dans le dossier M2105-IHMs**,

- créez un répertoire nommé **tp6**

**Documentation wxWidgets complète** : à installer **en local**, sur c:\temp\ par exemple, et pas sur un disque réseau.

#### A.- Programme graphique minimal : zero.exe

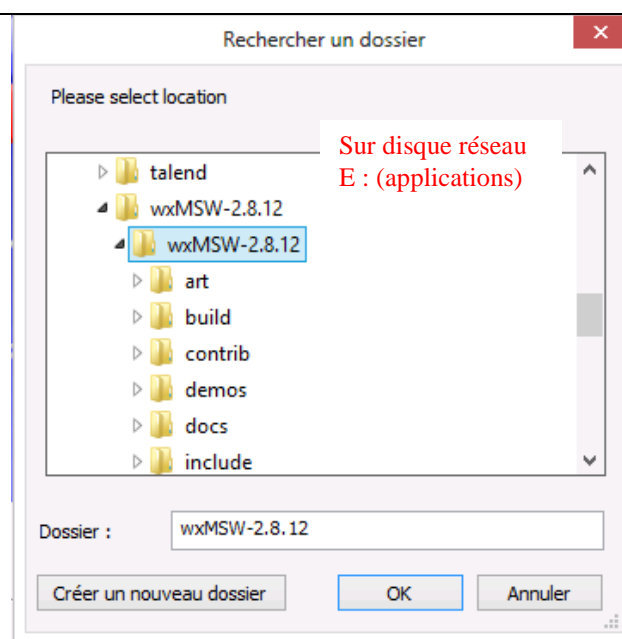
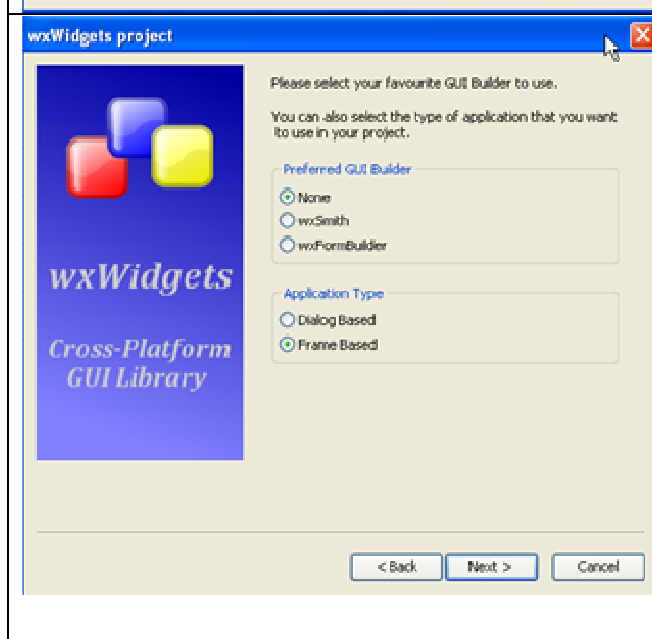
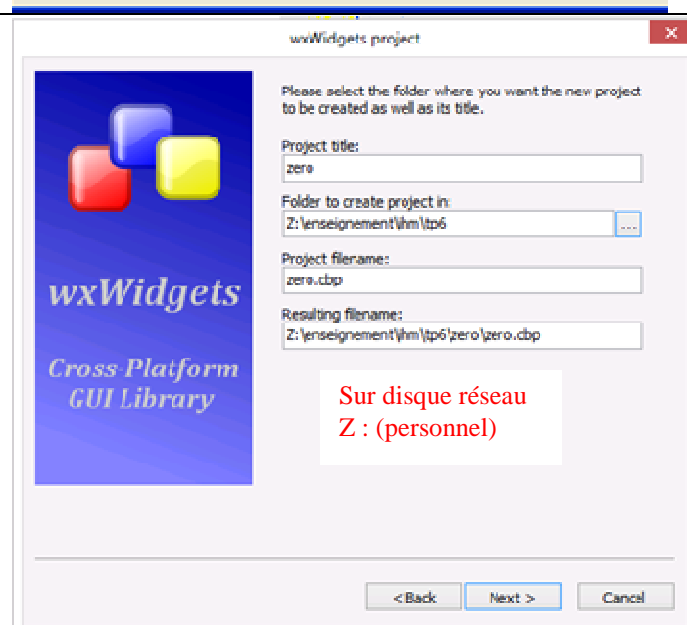
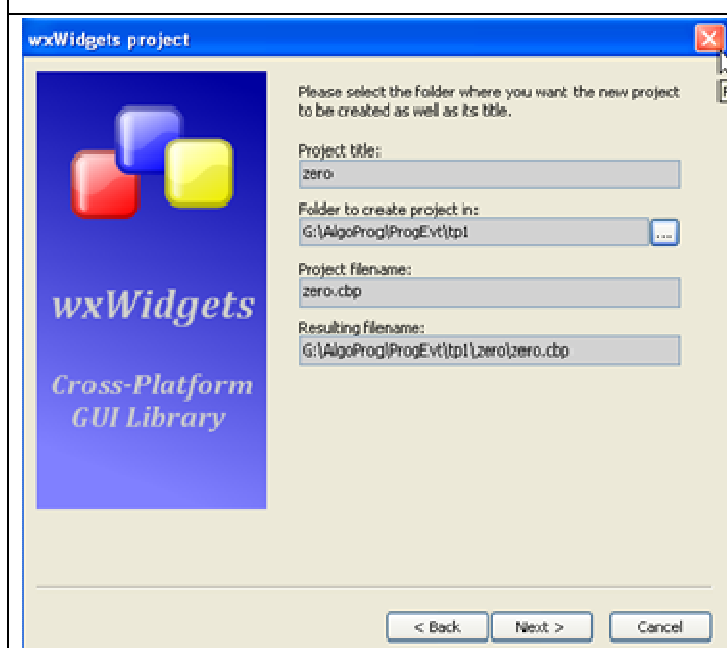
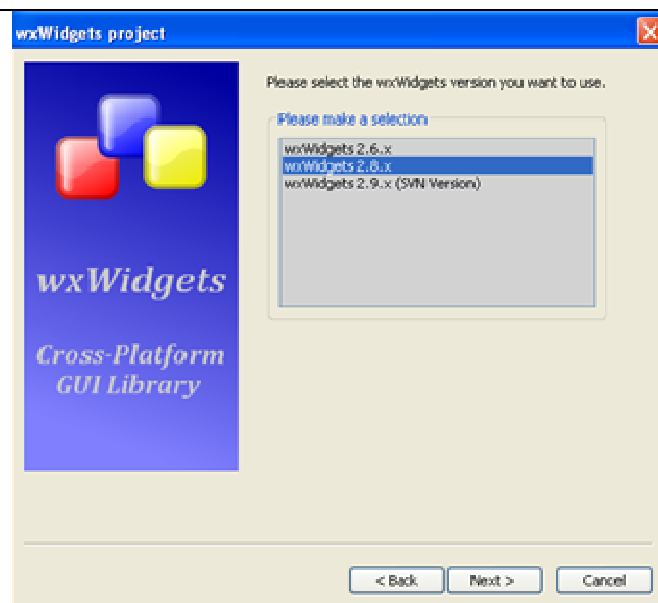
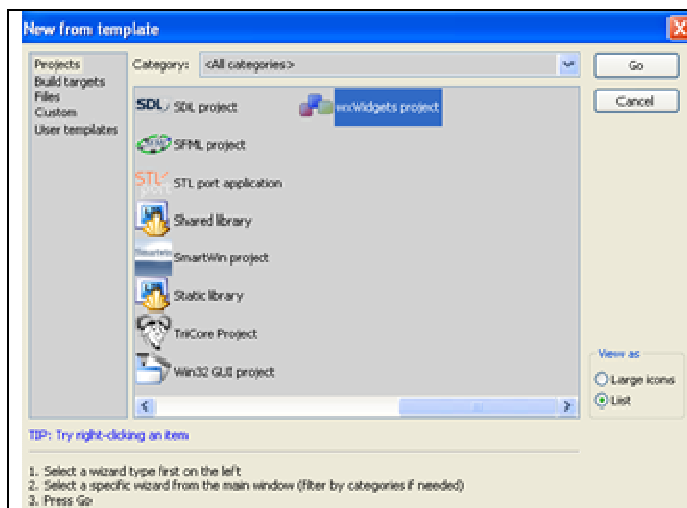
- (a) Dans votre répertoire tp6, **créez** un projet CodeBlock intitulé **zero** avec les options ci-dessous :
- template wxWidgets, version 2.8.x
  - **Projet SANS générateur d'interface GUI**, Application de type FRAMED
  - Compilateur par défaut Gnu-gcc, mode **DEBUG uniquement**
  - Bibliothèques **STATIQUES**, codage des caractères en mode **UNICODE** (meilleure portabilités des applications),
  - **Création d'un projet VIDE** (vous associerez au projet les sources mis à votre disposition une fois le projet vide créé)
  - Application sera en mode **GUI** (et non en mode CONSOLE)
  - Associer les bibliothèques **\_WXDEBUG\_** et **Debug wxWidgets** au compilateur Gnu-gcc
  - Application cible (type de l'exécutable .exe à fabriquer) : de type **GUI**

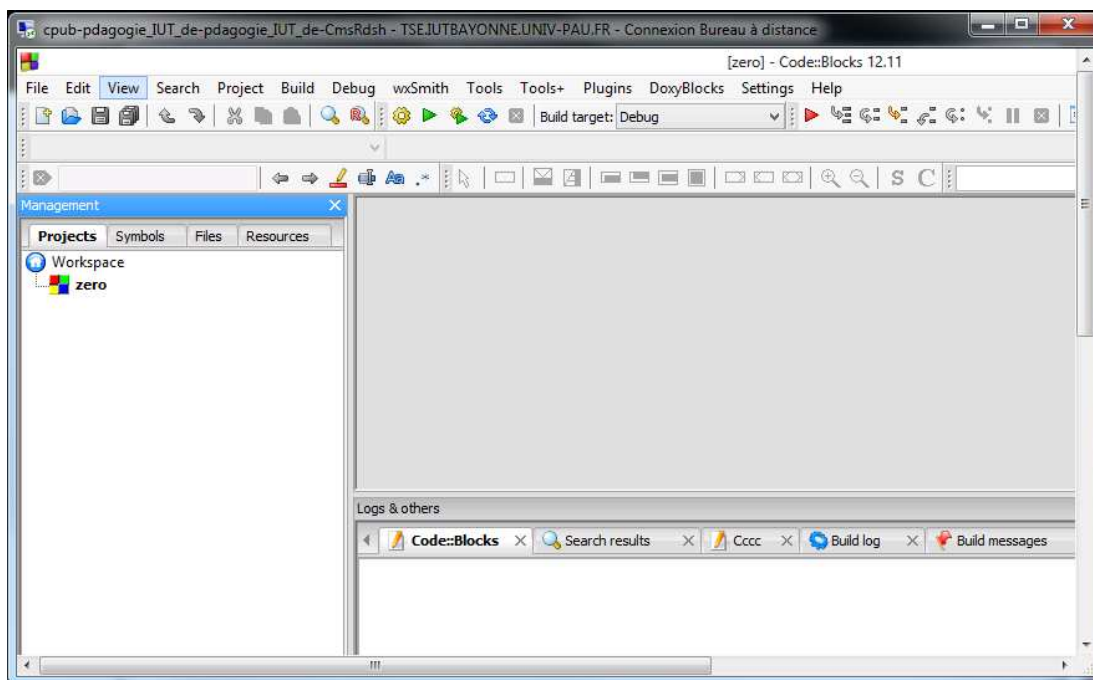
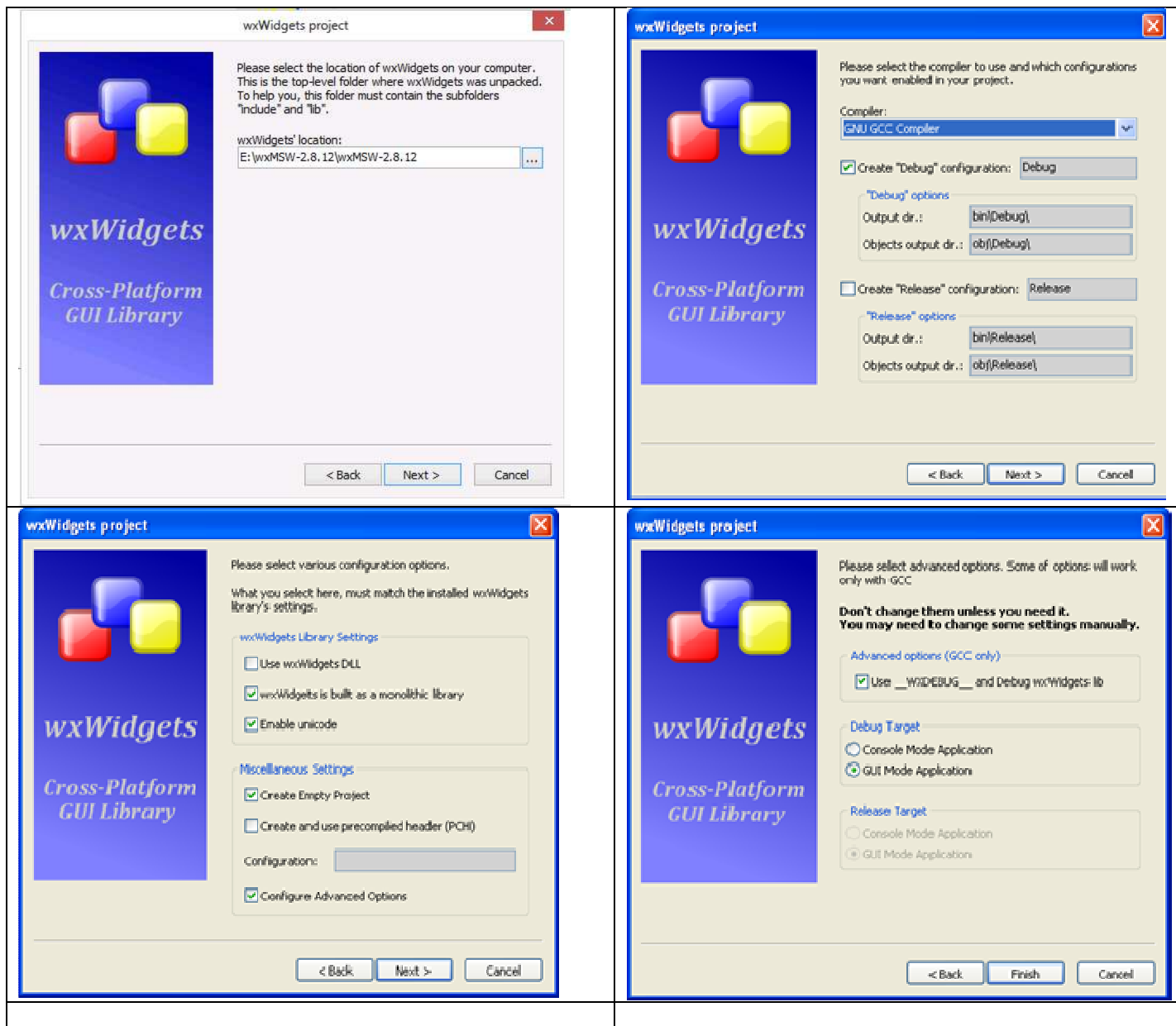
##### Résultat attendu :

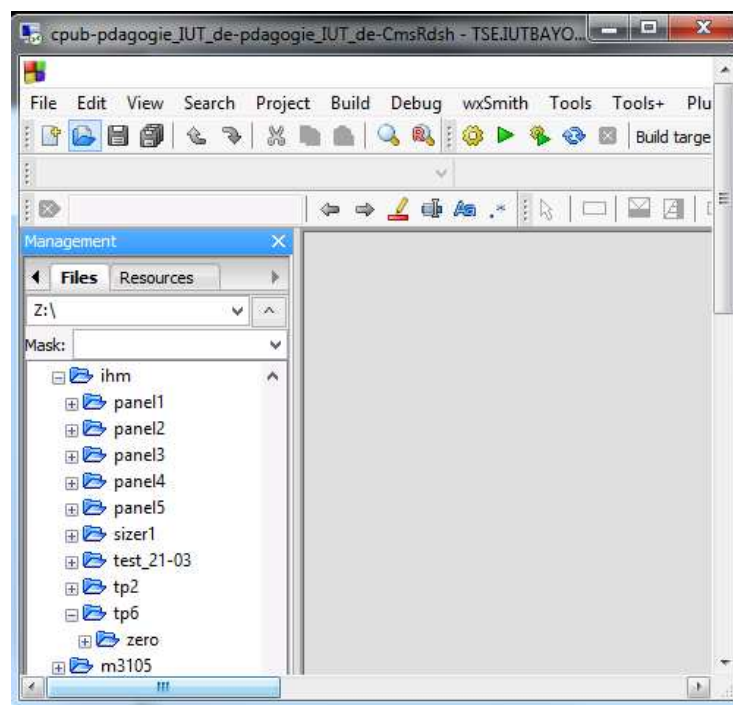
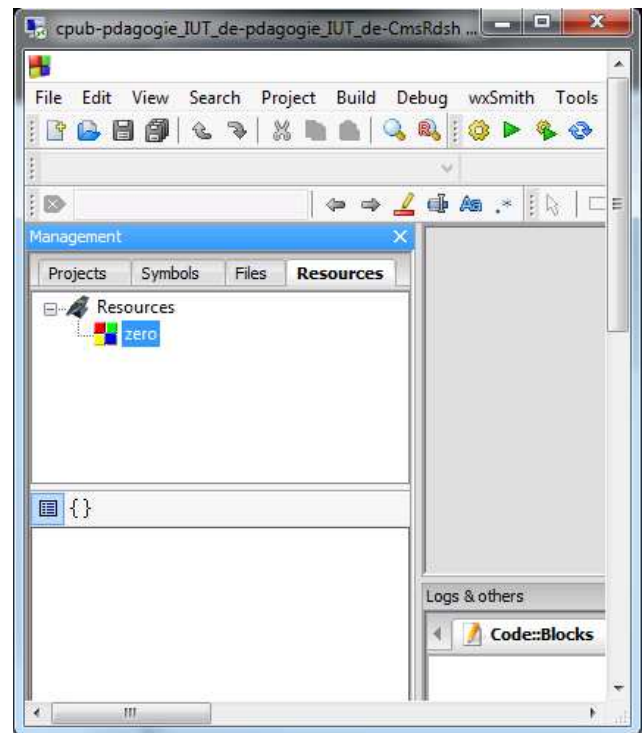
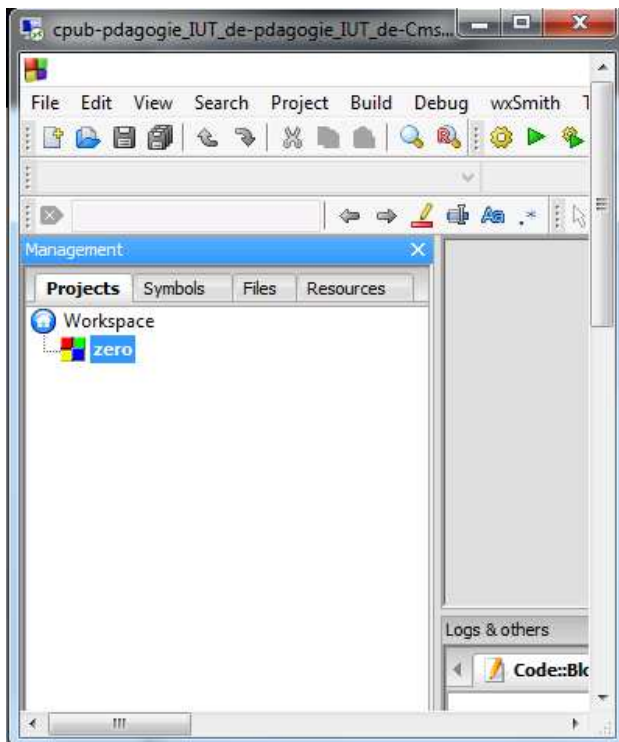
Un projet nommé zero, vide (sans fichier source attaché) et sans générateur d'interface associé a été créé dans votre répertoire \tp6 .

Si vous n'obtenez pas le résultat escompté, reportez-vous à la série de copies d'écran ci-dessous qui illustrent la démarche.

- (b) Dans le répertoire **tp5\zero**, **copiez** les 4 fichiers sources C++ contenus dans l'archive **zero.zip** mise à disposition dans la rubrique 'Ressources' de l'espace WebCampus du module.
- (c) **Compilez, construisez, exécutez.** Vous avez construit un programme Windows minimal. La fenêtre est déplaçable, redimensionnable. La fermeture de la fenêtre entraîne l'arrêt de l'application (vérifiez-le en consultant le Gestionnaire de tâches Windows).
- (d) Dans quel répertoire se trouve le fichier exécutable **zero.exe** créé ?

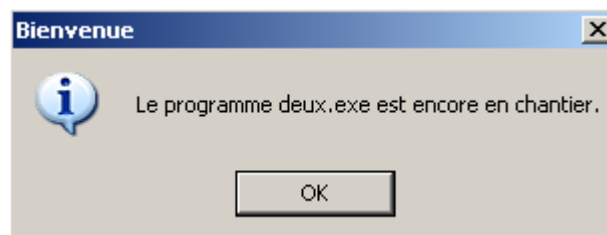






## B.- deux.exe : Additionneur simplifié (Première partie)

- (a) Dans votre répertoire tp6, créez un projet CodeBlock intitulé **deux** avec les mêmes options que celles du projet zero.
- (b) Dans le répertoire `tp6\deux`, **copiez** les 4 fichiers sources C++ contenus dans l'archive `zero.zip` mise à disposition dans la rubrique 'Ressources' de l'espace WebCampus du module.
- (c) Faites les modifications nécessaires dans les codes source de sorte à renommer les fichiers de la façon suivante : `appli2.h`, `appli2.cpp`, `principale2.h` et `principale2.cpp`
- (d) Associez ces fichiers source à votre projet **deux**
- (e) Mettre en conformité les noms des classes (`Principale2`, `Appli2`) et des variables `#define`
- (f) Donner à la barre de titre le libellé suivant : 'Principale 2 : Addition'
- (g) Modifier/Compléter les codes sources en rajoutant les lignes nécessaires à la **création** des éléments graphiques décrits dans la feuille de td n°5.
- (h) Message de bienvenue  
Au démarrage du programme, un message de bienvenue sera affiché au moyen d'une boîte de message, conformément à la copie d'écran ci-dessous.



- Consulter la documentation qui vous a été fournie (*Dialog functions*).
- Précisez le terme *function* utilisé : s'agit-il ou pas d'une méthode ? Si oui, de quelle classe ? Si non, expliquez.