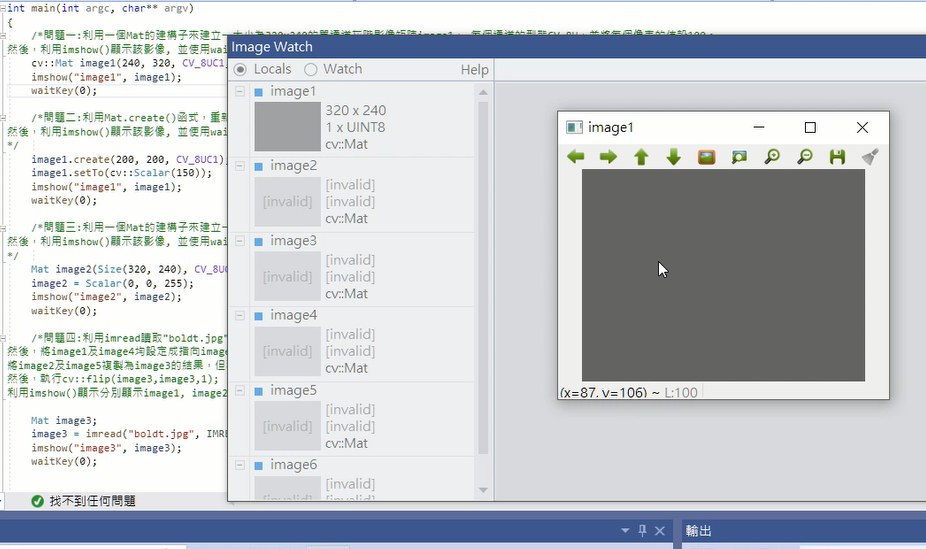
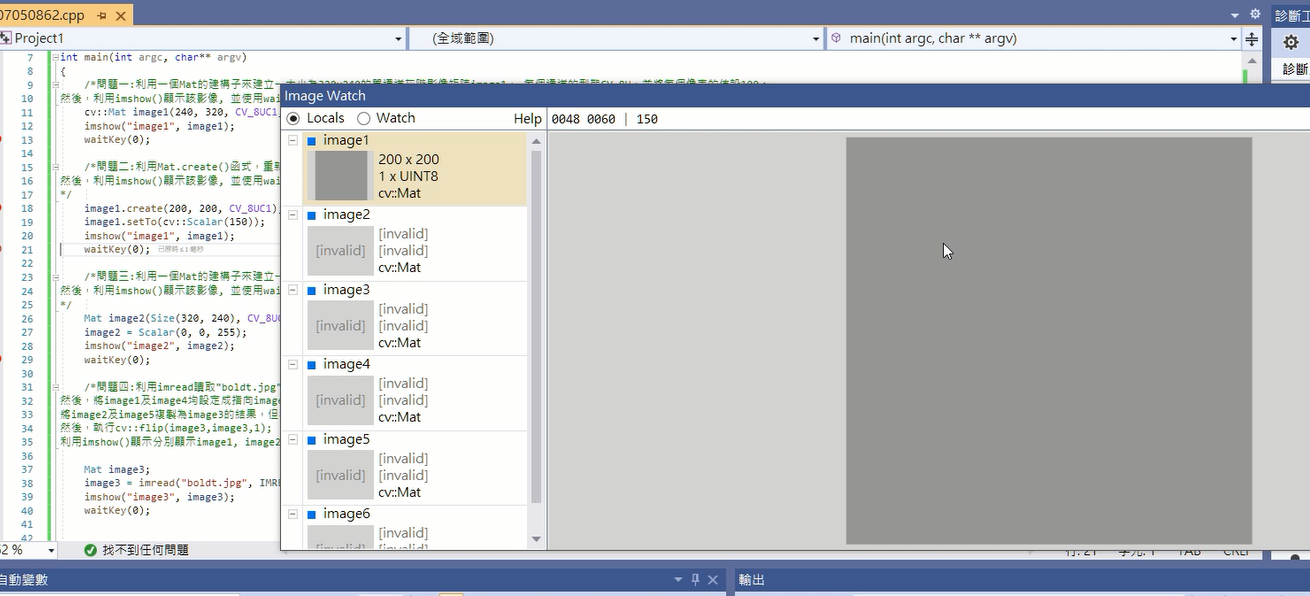
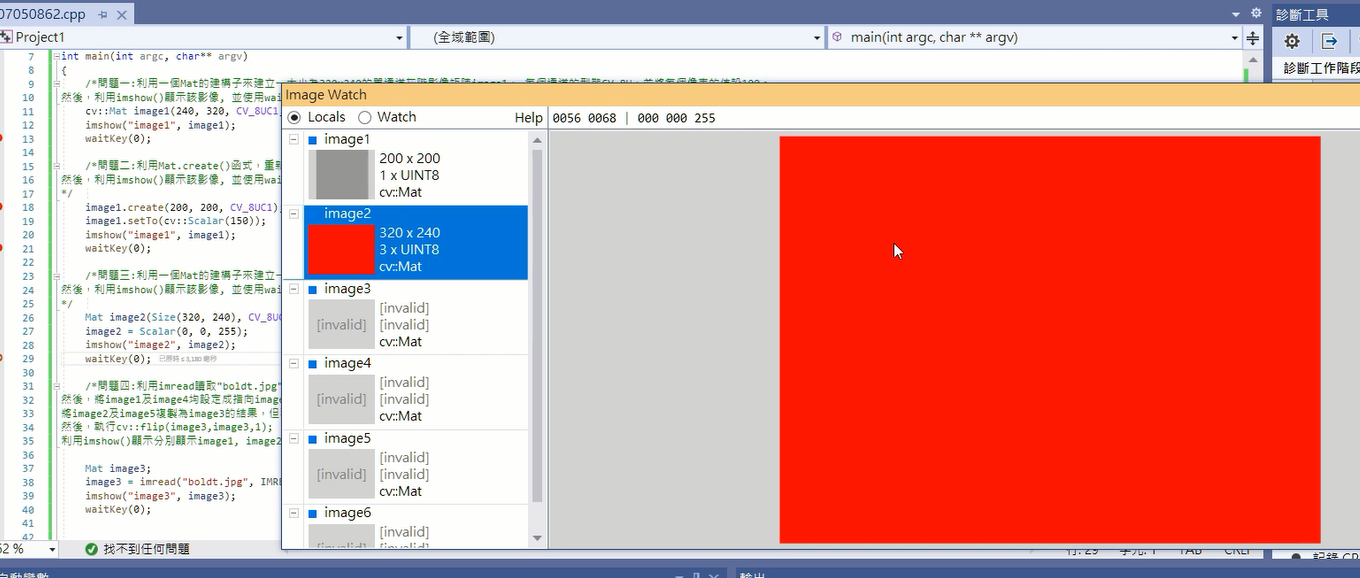
**問題1結果**



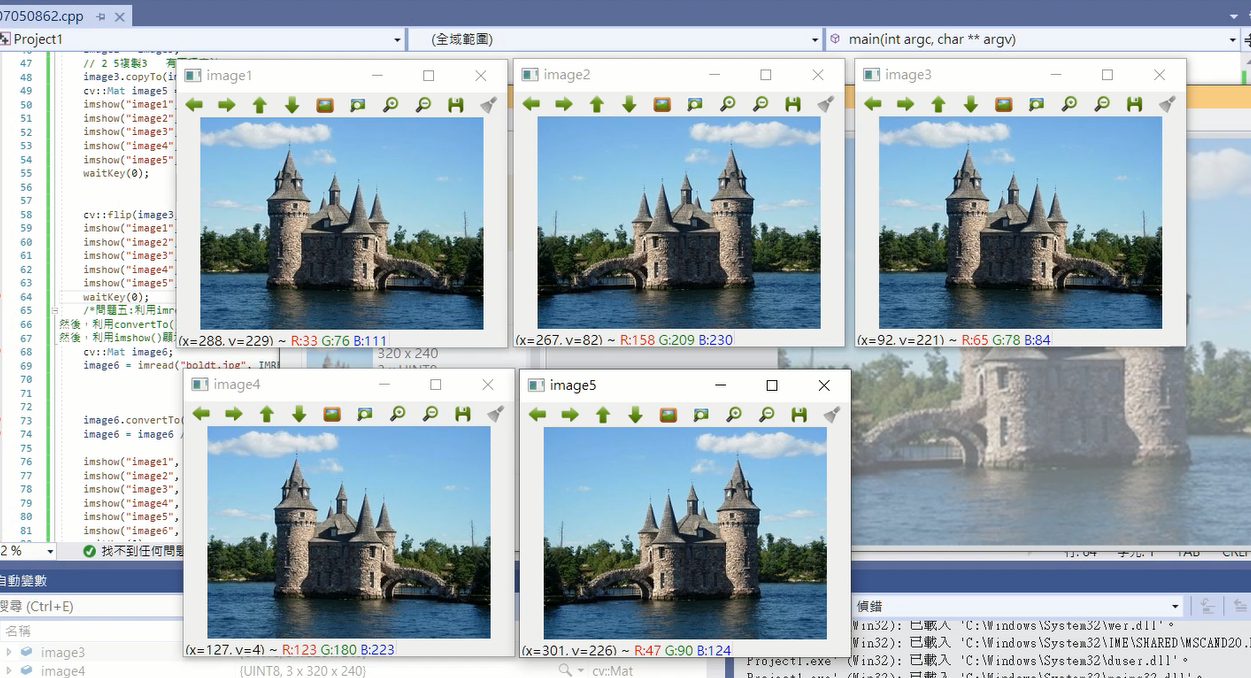
**問題2結果**



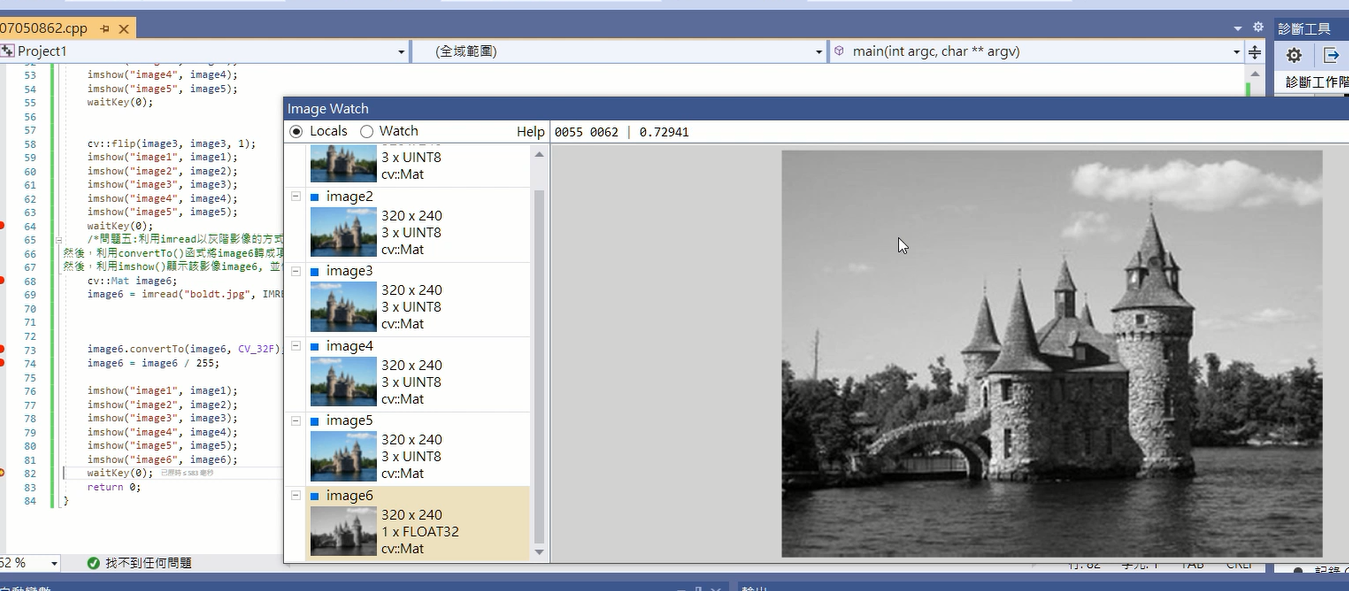
**問題3結果**



**問題4結果**



**問題5結果**



|  |
| --- |
| #include <iostream>  #include <opencv2/opencv.hpp>  using namespace std;  using namespace cv;  int main(int argc, char\*\* argv)  {  /\*問題一:利用一個Mat的建構子來建立一大小為320x240的單通道灰階影像矩陣image1， 每個通道的型態CV\_8U，並將每個像素的值設100。  然後，利用imshow()顯示該影像, 並使用waitKey(0)等待按任意鍵。\*/  cv::Mat image1(240, 320, CV\_8UC1, Scalar(100));  imshow("image1", image1);  waitKey(0);  /\*問題二:利用Mat.create()函式，重新配置image1，使其大小為200x200。再將所有像素的值設150 。  然後，利用imshow()顯示該影像, 並使用waitKey(0)等待按任意鍵。  \*/  image1.create(200, 200, CV\_8UC1);  image1.setTo(cv::Scalar(150));  imshow("image1", image1);  waitKey(0);  /\*問題三:利用一個Mat的建構子來建立一大小為320x240的三通道彩色影像矩陣image2， 每個通道的型態CV\_8U，通道的順序為BGR，並將每個像素的顏色設純紅色。  然後，利用imshow()顯示該影像, 並使用waitKey(0)等待按任意鍵。  \*/  Mat image2(Size(320, 240), CV\_8UC3);  image2 = Scalar(0, 0, 255);  imshow("image2", image2);  waitKey(0);  /\*問題四:利用imread讀取"boldt.jpg" 存為image3。  然後，將image1及image4均設定成指向image3的資料區塊。  將image2及image5複製為image3的結果，但獨立於image3的資料區塊。  然後，執行cv::flip(image3,image3,1);  利用imshow()顯示分別顯示image1, image2, image3, image4及image5，觀察它們之間的差別, 並使用waitKey(0)等待按任意鍵。\*/  Mat image3;  image3 = imread("boldt.jpg", IMREAD\_COLOR);  imshow("image3", image3);  waitKey(0);  // 1 4指向3 有兩種寫法  //Mat image4, image5;  cv::Mat image4(image3);  image1 = image3;  // 2 5複製3 有兩種寫法  image3.copyTo(image2);  cv::Mat image5 = image3.clone();  imshow("image1", image1);  imshow("image2", image2);  imshow("image3", image3);  imshow("image4", image4);  imshow("image5", image5);  waitKey(0);  cv::flip(image3, image3, 1);  imshow("image1", image1);  imshow("image2", image2);  imshow("image3", image3);  imshow("image4", image4);  imshow("image5", image5);  waitKey(0);  /\*問題五:利用imread以灰階影像的方式讀取"boldt.jpg" 存為image6。  然後，利用convertTo()函式將image6轉成項目型態為CV\_32F的浮點數矩陣，使每個像素值介於[0,1]的範圍。  然後，利用imshow()顯示該影像image6, 並使用waitKey(0)等待按任意鍵。\*/  cv::Mat image6;  image6 = imread("boldt.jpg", IMREAD\_GRAYSCALE);    image6.convertTo(image6, CV\_32F);  image6 = image6 / 255;    imshow("image1", image1);  imshow("image2", image2);  imshow("image3", image3);  imshow("image4", image4);  imshow("image5", image5);  imshow("image6", image6);  waitKey(0);  return 0;  } |