學號:41247001S

輸出的結果和我原本想的不太一樣。不過後來查document之後發現原因了。

uint32_t 的 -1 = 4294967295 (2倍的2^32- 1)
這是因為 int32_t 儲存 - 1的方式在二進位制
跟uint32_t 的 4294967295 一樣所導致的

當 uint32_t + int32_t 時，運算上面編譯會把int32_t先轉換成 uint32_t，因此就會變成 0 + 4294967295 = 4294967295

而為甚麼當 uint16_t + int32_t 時就不會呢？
這是因為 int32_t 所使用的記憶體比uint16_t大。
（兩個變數的記憶體大小不同）
所以這時編譯就會變成把 uint16_t 轉成 int32_t。
變成 0 + (-1) = -1 。
而不會像前面的uint32_t 一樣變成 4294967295

此外，我也有找到會自動發生強制轉換的情形。例如兩者的資料型態不同、兩者所使用的記憶體大小不同等。