vc-3. ダンプリスト，配列

（Visual Studio C++ の機能と操作）

https://www.kkaneko.jp/cc/vc/index.html

金子邦彦

目次

3-1 ダンプリスト

メモリとアドレス

- **メモリ**は**バイト**（８ビット）単位に区切られている

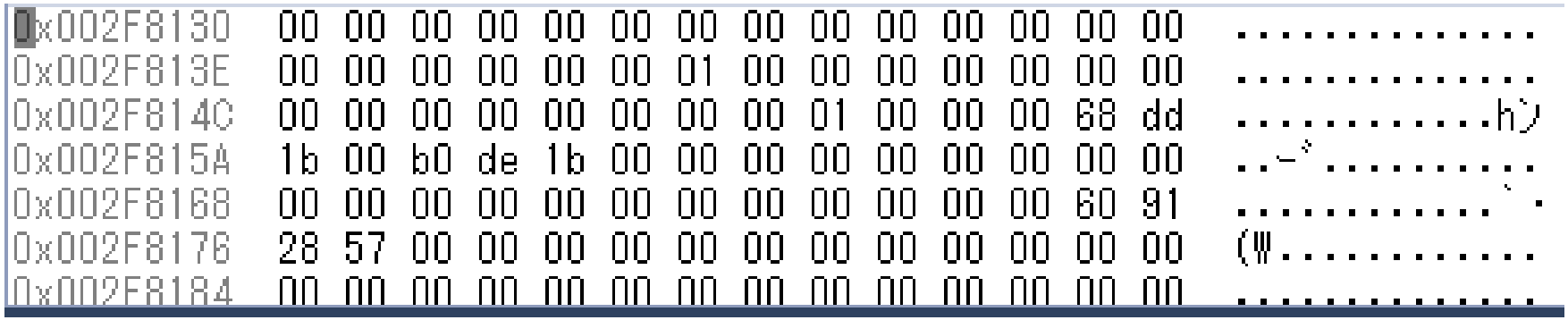
- 各**バイト**には０から始まる通し番号が付けられている。これが**アドレス**（番地ともいう）

メモリ内のデータは

01	00	00	00	02	00	00	00	03	00
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

アドレス

ダンプリストの例



```
0x002F8130  00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..............
0x002F813E  00 00 00 00 00 00 01 00 00 00 00 00 00 00  ..............
0x002F814C  00 00 00 00 00 00 00 00 01 00 00 00 68 dd  ............hﾝ
0x002F815A  1b 00 b0 de 1b 00 00 00 00 00 00 00 00 00  ..ｰﾟ..........
0x002F8168  00 00 00 00 00 00 00 00 00 00 00 00 60 91  ............`'
0x002F8176  28 57 00 00 00 00 00 00 00 00 00 00 00 00  (W............
0x002F8184  00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..............
```

アドレス

メモリの中身

バイト単位で区切られて表示.

表示は**16進数**

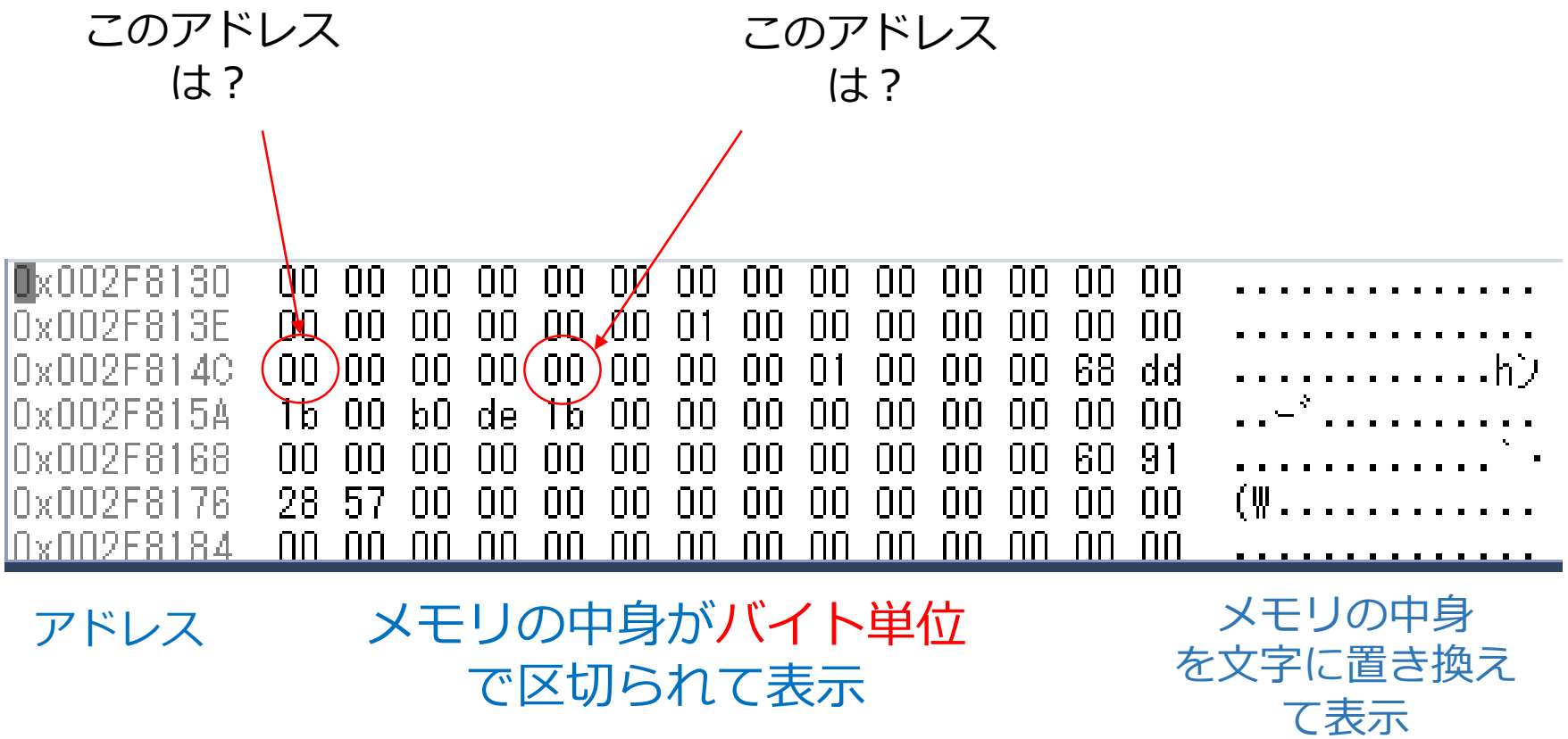
ダンプリストの例

アドレスは
0x002f8130

アドレスは
0x002f8131

アドレスは
0x002f8132

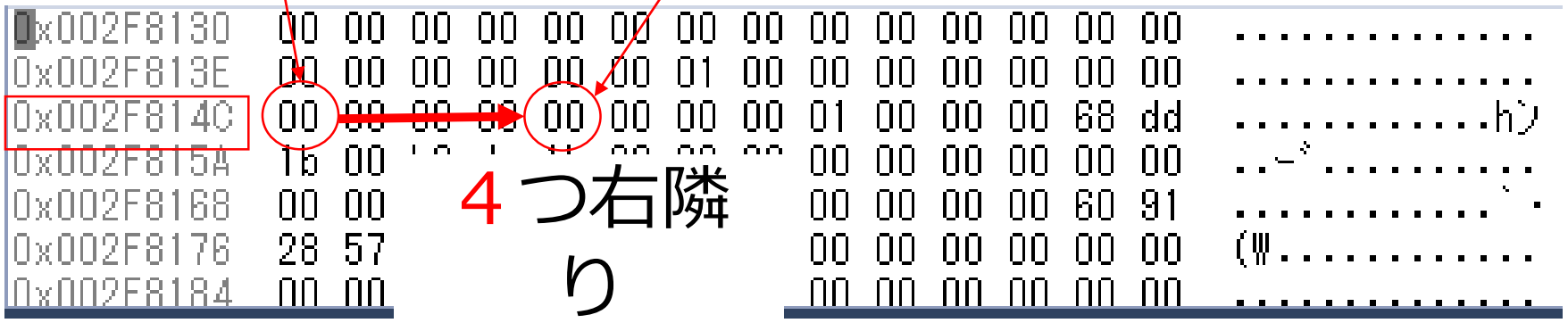
クイズ


このアドレスは？
このアドレスは？
0x002F8130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
0x002F813E 00 00 00 00 00 00 01 00 00 00 00 00 00 00 ....
0x002F814C 00 00 00 00 00 00 00 00 01 00 00 00 68 dd ....hﾝ
0x002F815A 1b 00 b0 de 1b 00 00 00 00 00 00 00 00 00 ..ｰﾞ..........
0x002F8168 00 00 00 00 00 00 00 00 00 00 00 00 60 91 ....`.
0x002F8176 28 57 00 00 00 00 00 00 00 00 00 00 00 00 (W............
0x002F8184 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
アドレス
メモリの中身がバイト単位で区切られて表示
メモリの中身を文字に置き換えて表示

クイズの答え

0x002F814C
左のアドレス
表示で分かる

0x002F8150
これは 0x002F814C に 4 足した値



4 つ右隣り

アドレス

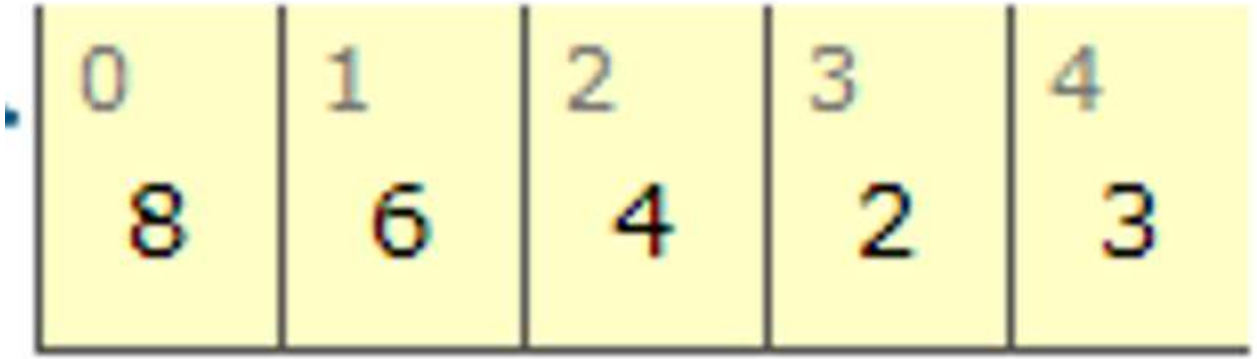
メモリの中身がバイト単位で区切られて表示

メモリの中身を文字に置き換えて表示

3-2 配列

3-2 配列

- **配列**は、**同じ型の要素の並び**.
- ０から始まる番号（**添字**）が付いている



- **コード化**されて、**メモリ**に格納されるとき、

  要素が**順に**メモリに**格納される**

  各要素の**サイズは同じ**

配列

４バイトの数値に**コード化**した場合

C/C++ での配列と繰り返し

```
int main()
{
    static int x[5] = { 8, 6, 4, 2, 3 };
    static int y[5] = { 0, 0, 0, 0, 0 };
    int i;
    for (i = 0; i < 5; i++) {
        y[i] = x[i] * 10;
    }

    return 0;
}
```

繰り返す処理

i の値は 0, 1, 2, 3, 4
と変化し，全部済んだら終わる

パソコン演習

- Visual Studio 2015 を起動しなさい
- Visual Studio 2015 で、Win32 コンソールアプリケーション用プロジェクトを新規作成しなさい

プロジェクトの「名前」は何でもよい

- Visual Studio 2015 のエディタを使って、ソースファイルを編集しなさい

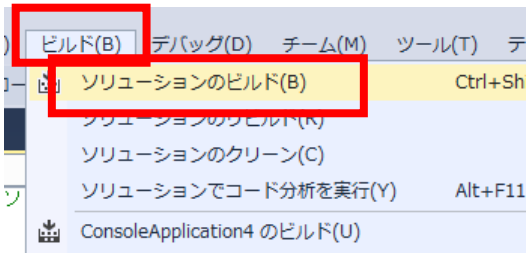
```
#include "stdafx.h"

int main()
{
    static int x[5] = { 8, 6, 4, 2, 3 };
    static int y[5] = { 0, 0, 0, 0, 0 };
    int i;
    for (i = 0; i < 5; i++) {
        y[i] = x[i] * 10;
    }

    return 0;
}
```

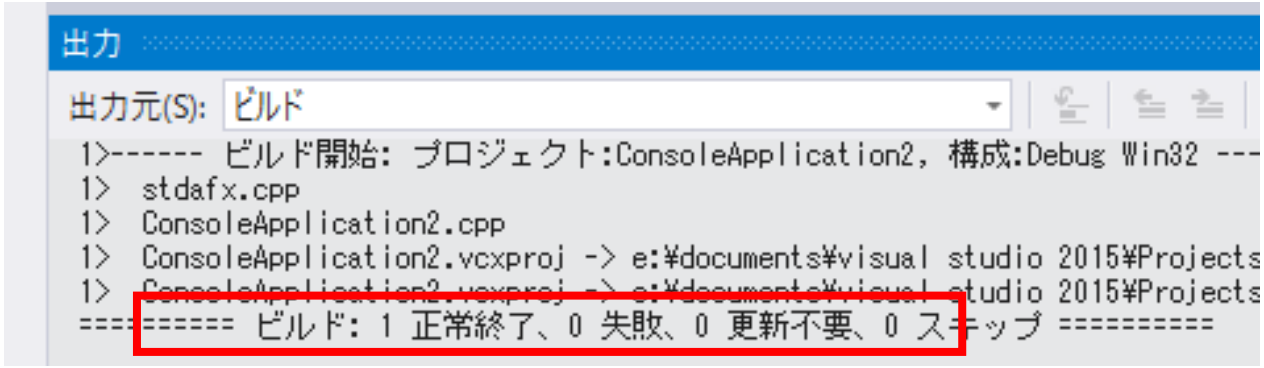
６行追加

```
#include "stdafx.h"


int main()
{
    static int x[5] = { 8, 6, 4, 2, 3 };
    static int y[5] = { 0, 0, 0, 0, 0 };
    int i;
    for (i = 0; i < 5; i++) {
        y[i] = x[i] * 10;
    }

    return 0;
}
```

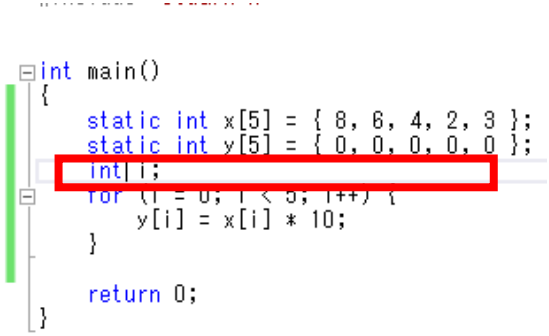
- ビルドしなさい．ビルドのあと「１　正常終了、０　失敗」の表示を確認しなさい

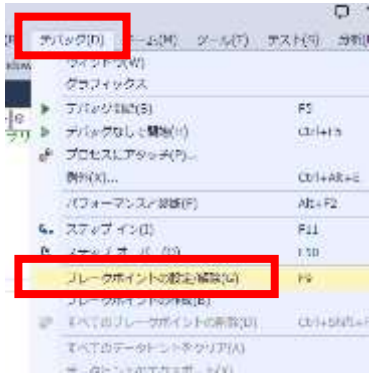
→　表示されなければ、プログラムのミスを自分で確認し、修正して、ビルドをやり直す

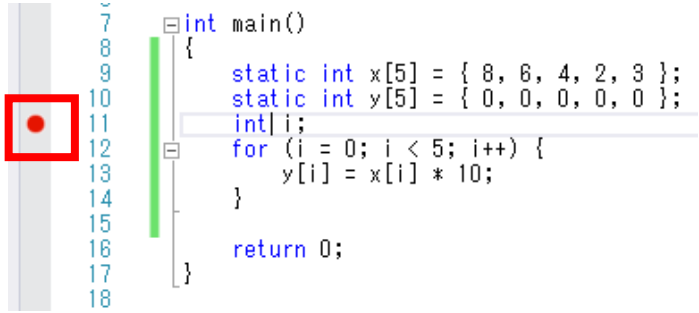




- Visual Studio 2015 で「int i」の行に、ブレークポイントを設定しなさい





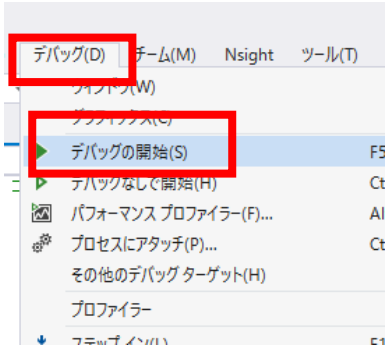


① 「**int i;**」の行をマウスでクリック

② 「デバッグ」→「**ブレークポイントの設定/解除**」

③ ブレークポイントが設定されるので確認．赤丸がブレークポイントの印

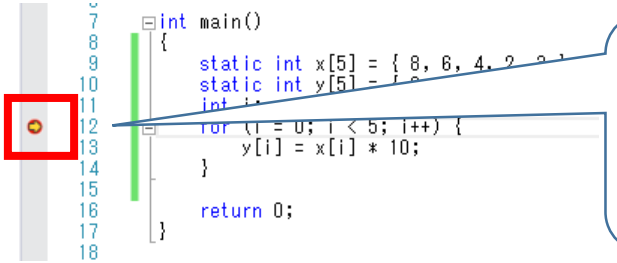
- Visual Studio 2015 で、デバッガーを起動しなさい.



「デバッグ」
→「**デバッグの開始**」

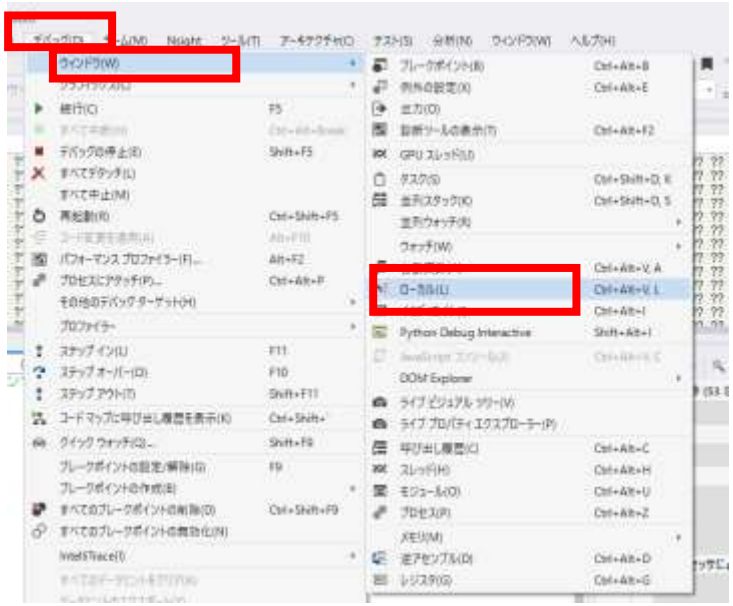
- 「int i;」の行で、実行が中断することを確認しなさい

あとで使うので、中断したままにしておくこと

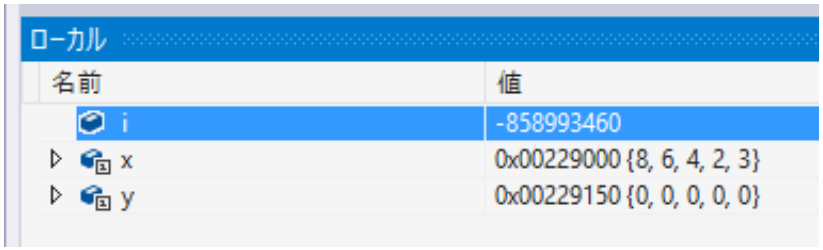


「int i;」の行で実行が中断している

- 「int i;」の行で、実行が中断した状態で、変数の値を表示させなさい．手順は次の通り．



①「デバッグ」
→「ウインドウ」
→ 「**ローカル**」



② 変数名と値の対応表が表示される

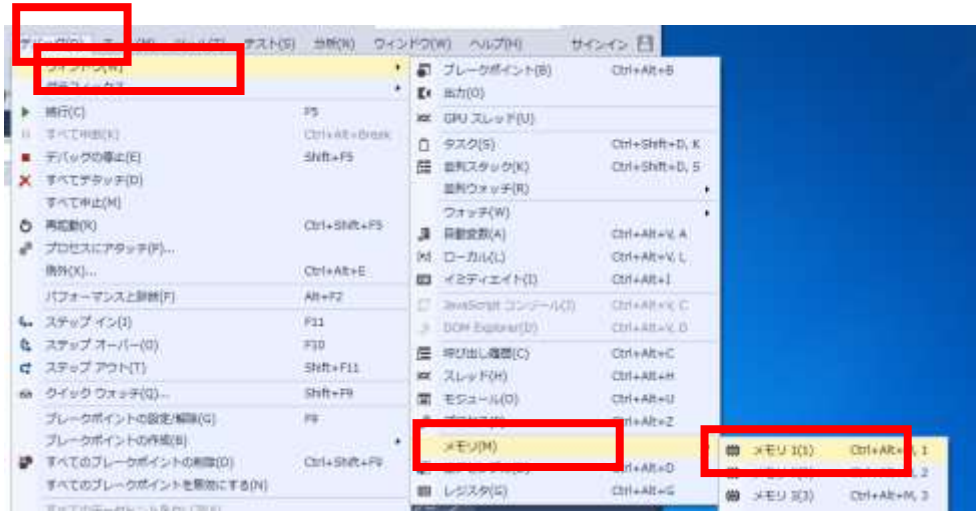
※ 次ページに拡大図

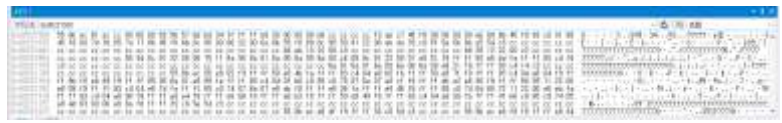
9. **ローカルウインドウ**で**配列 y の先頭アドレス**を調べなさい

「0x」が付いているのは16進数

y の先頭アドレスは、起動のたびに
変化する可能性がある.

- メモリの中身を表示させなさい．手順は次の通り．

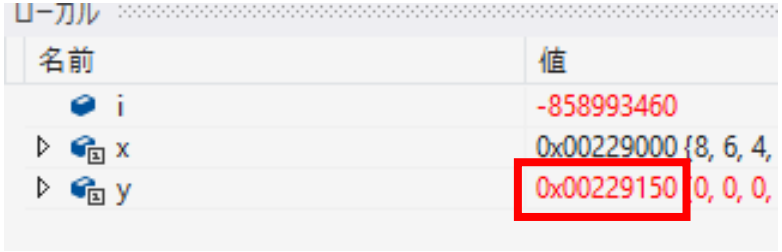




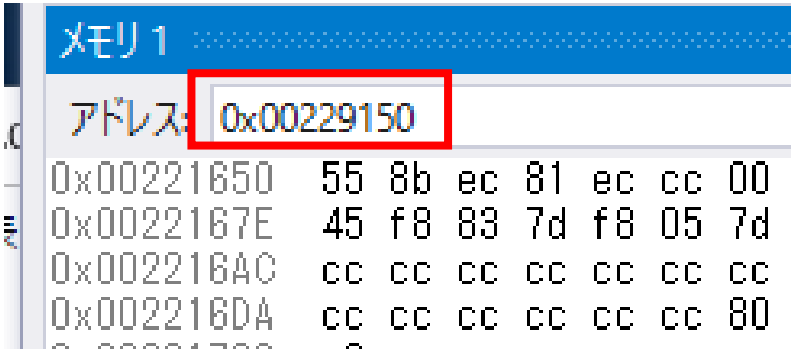
② メモリの中身がダンプリスト形式で表示される

①「デバッグ」
→「ウインドウ」
→ 「メモリ」→「メモリ１」

- **配列 y の先頭アドレス**を、
  メモリウインドウの**「アドレス」**のところに書き写して，
  Enter キーを押す

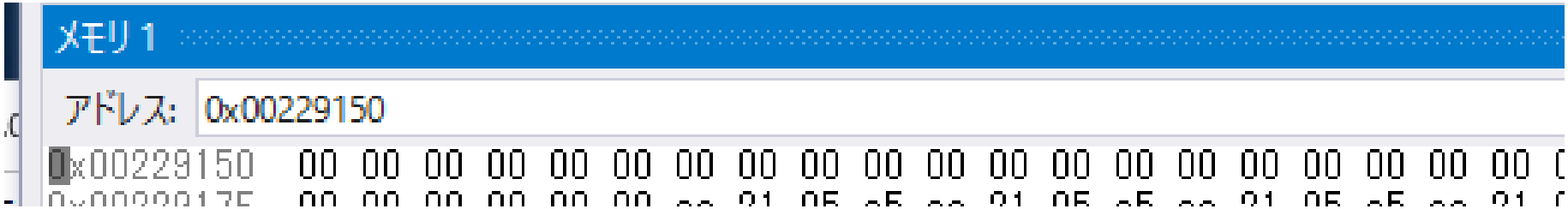


「0x00229150」のように
頭に 0x を付ける



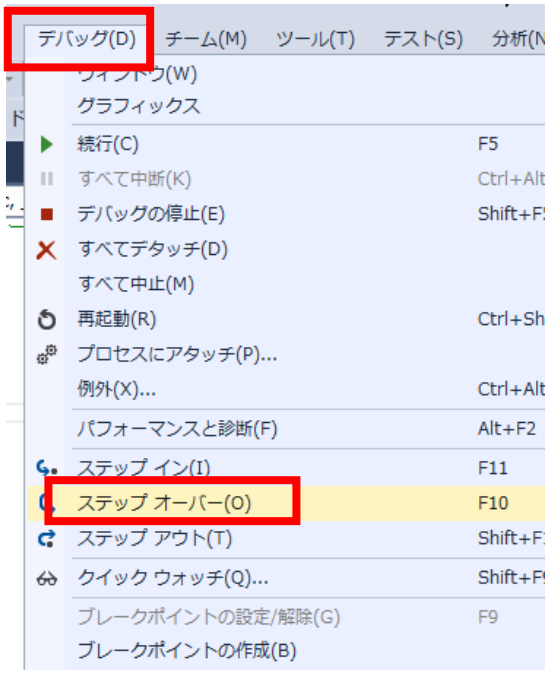
① 配列 y の先頭アドレス

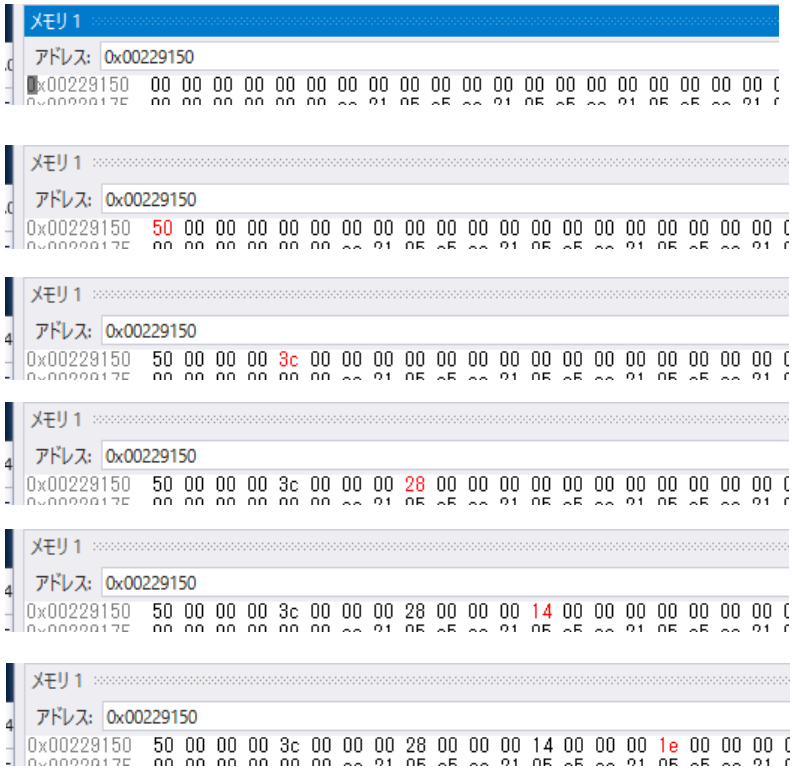
- **メモリウインドウに、配列 y の中身が表示されるので確認する**



00 が並んでいる

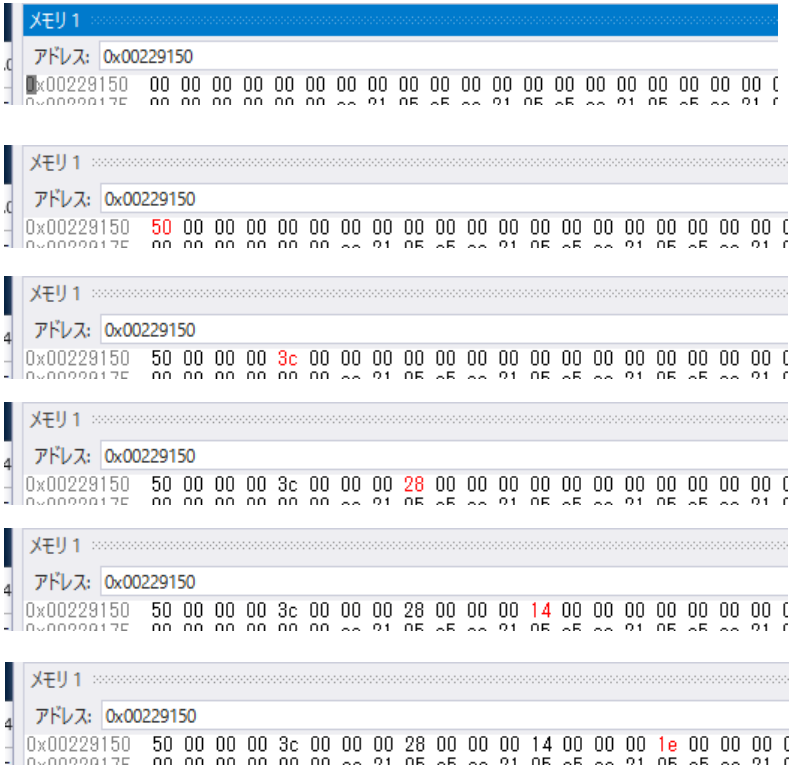
- ステップオーバーの操作を行いながら、メモリの中身の変化を確認しなさい.





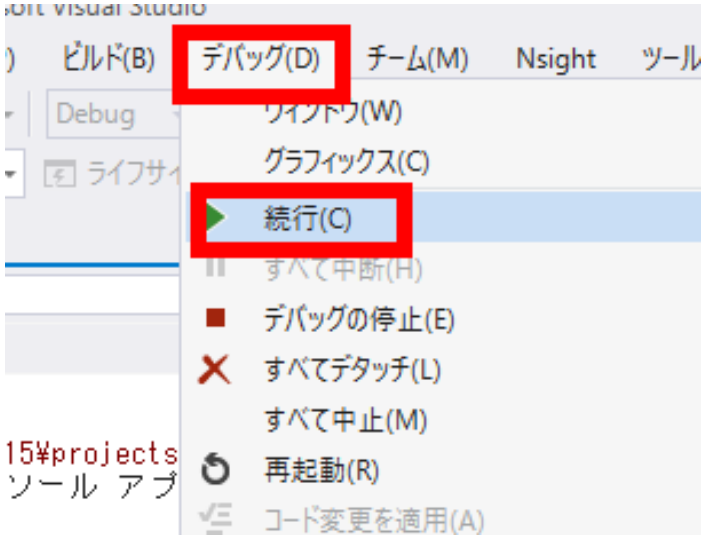
「デバッグ」
→「ステップオーバー」
（あるいは F10 キー）

- 次のことを再確認しなさい



16進数	10進数
50	**80**
3c	**60**
28	**40**
14	**20**
1e	**30**

- 最後に、プログラム実行の再開の操作を行いなさい。これで、デバッガーが終了する。



「デバッグ」
→「**続行**」