vc-4. 文字データと文字コード

（Visual Studio C++ の機能と操作）

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

金子邦彦

ASCII 文字コード表

コンピュータで

英数文字データを扱うときの標準

	0	1	2	3	4	5	6	7
0	**NULL**	**DEL**	**SP**	**0**	**@**	**P**		**p**
1	**SOH**	**DC1**	**!**	**1**	**A**	**Q**	**a**	**q**
2	**STX**	**DC2**	**“**	**2**	**B**	**R**	**b**	**r**
3	**ETX**	**DC3**	**#**	**3**	**C**	**S**	**c**	**s**
4	**EOT**	**DC4**	**$**	**4**	**D**	**T**	**d**	**t**
5	**ENQ**	**NAK**	**%**	**5**	**E**	**U**	**e**	**u**
6	**ACK**	**SYN**	**&**	**6**	**F**	**V**	**f**	**v**
7	**BEL**	**ETB**	**’**	**7**	**G**	**W**	**g**	**w**
8	**（BS）**	**CAN**	**（**	**8**	**H**	**X**	**h**	**x**
9	**（HT）**	**EM**	**）**	**9**	**I**	**Y**	**i**	**y**
A	**（LF）**	**SUB**	**＊**	**：**	**J**	**Z**	**j**	**z**
B	**（VT）**	**ESC**	**＋**	**；**	**K**	**［**	**k**	**{**
C	**（FF）**	**（FS）**	**，**	**＜**	**L**	**¥**	**l**	**|**
D	**（CR）**	**（GS）**	**－**	**＝**	**M**	**］**	**m**	**}**
E	**SO**	**（RS）**	**．**	**＞**	**N**	**＾**	**n**	**~**
F	**SI**	**（US）**	**／**	**？**	**O**	**＿**	**o**	**DEL**

太字は特別用途

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

```
 7  int main()
 8  {
 9      char message[] = "hello";
10      printf("%s", message);
11      return 0;
12  }
```

- 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

```
1>------ ビルド開始: プロジェクト:ConsoleApplication2, 構成:Debug Win32 ------
1>  ConsoleApplication2.cpp
1>  ConsoleApplication2.vcxproj -> e:¥Documents¥Visual Studio 2013¥Projects¥ConsoleApplication2¥Debug¥ConsoleApplication2.exe
========== ビルド: 1 正常終了、0 失敗、0 更新不要、0 スキップ ==========
```

- Visual Studio 2015 で「char message = “hello”;」の行に、ブレークポイントを設定しなさい

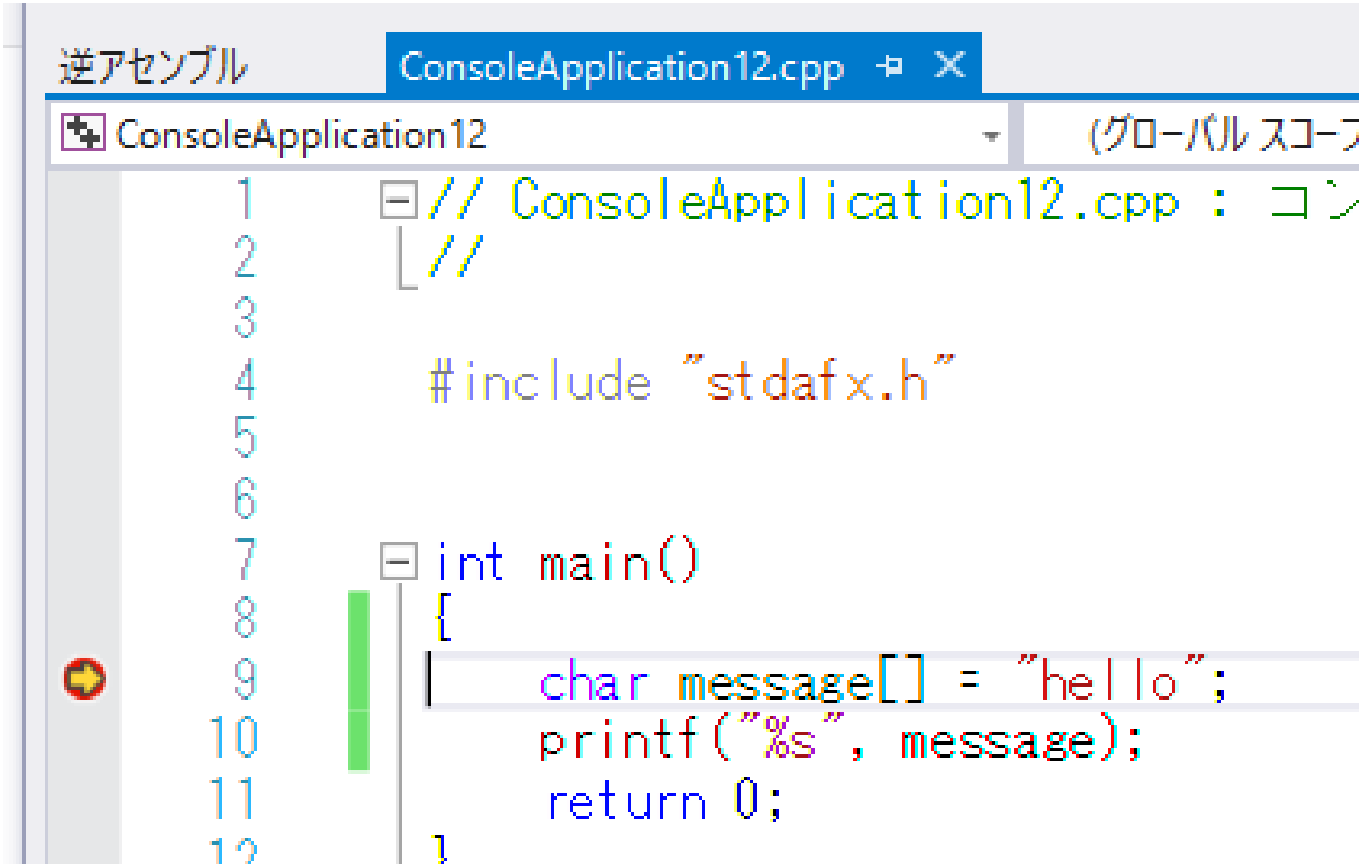
```
7   int main()
8   {
9       char message[] = "hello";
10      printf("%s", message);
11      return 0;
12  }
```

- Visual Studio 2015 で、デバッガーを起動しなさい.

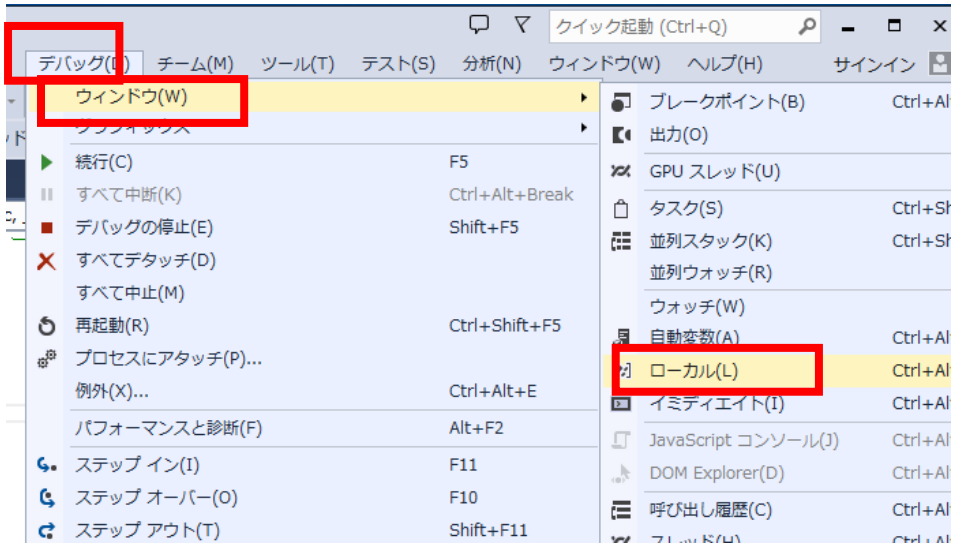
「デバッグ」
→「デバッグ開始」

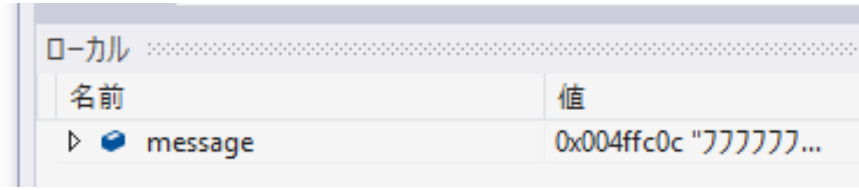
- 「char message = “hello”;」の行で、実行が中断することを確認しなさい

あとで使うので、中断したままにしておくこと



- 「 char message = “hello”;」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．

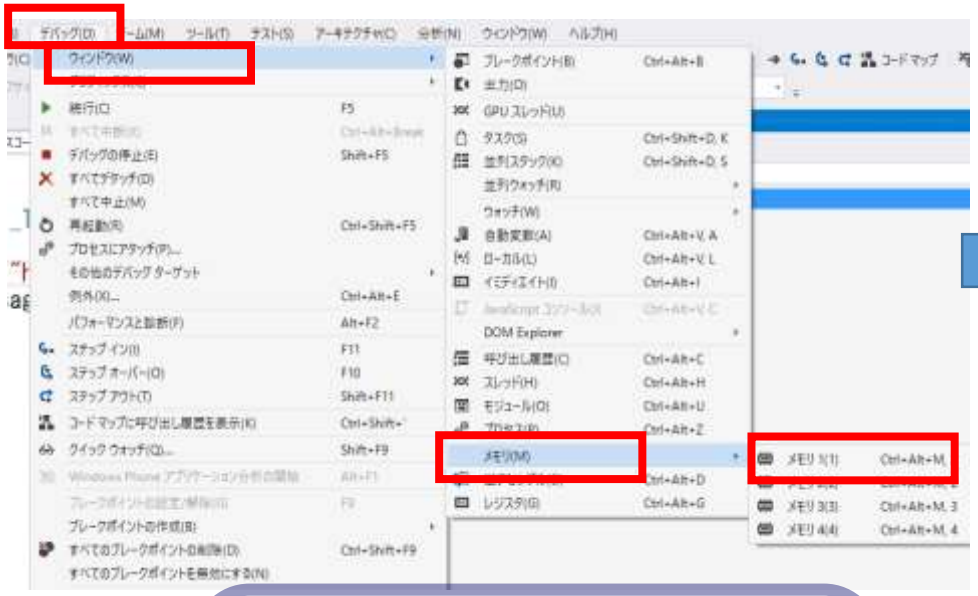


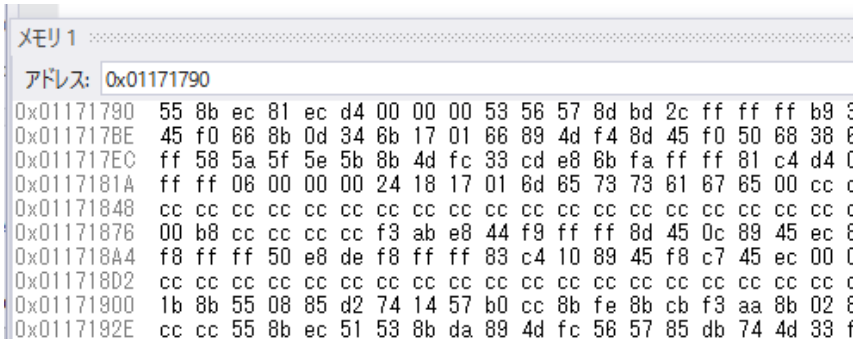


② 変数名と値の対応表が表示される

①「デバッグ」
→「ウインドウ」
→ 「**ローカル**」

- 「 char message = “hello”;」の行で、実行が中断した状態で、メモリの中身を表示させなさい．手順は次の通り．

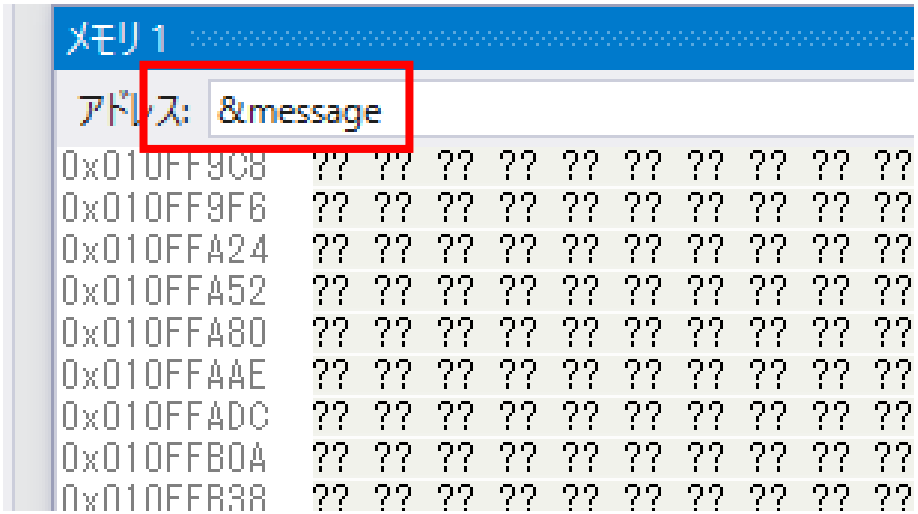


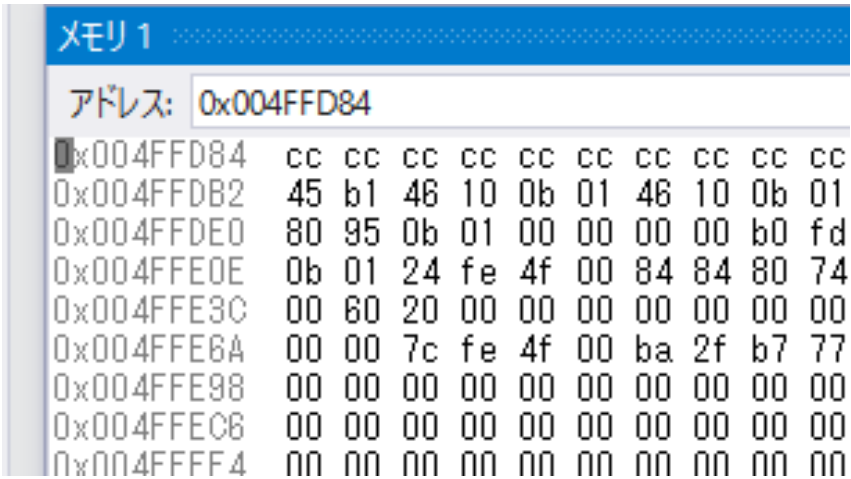


② 「メモリ１」の画面が表示される

①「デバッグ」
→「ウインドウ」
→ 「**メモリ**」
→ 「**メモリ１（１）**」

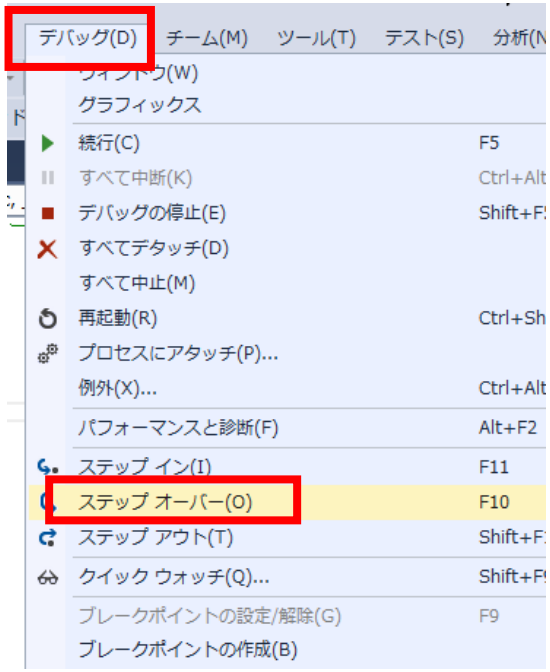
- 「メモリ１」の画面の「アドレス」に
「&message」と入れて Enter キーを押しなさい



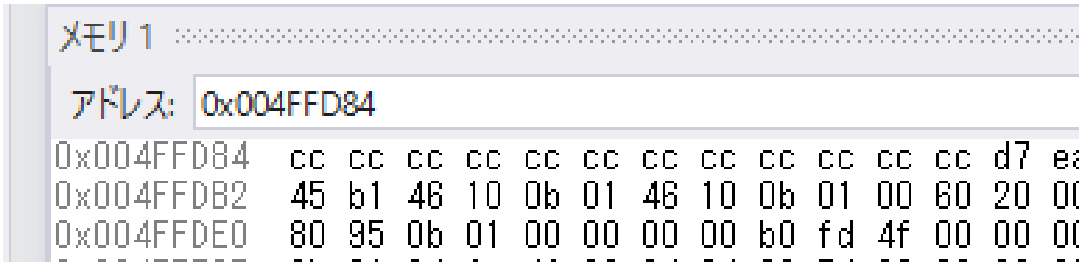


画面が変化するので確認する

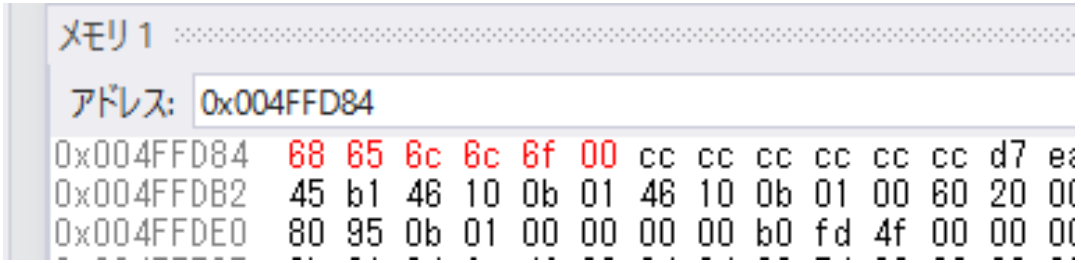
- ステップオーバーの操作を1回行いなさい

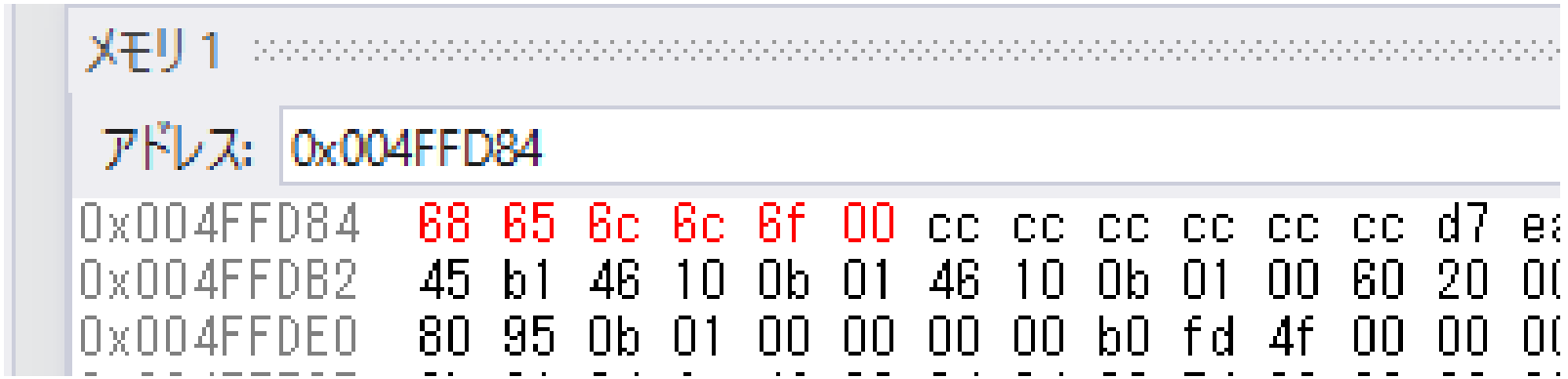


「デバッグ」
→「ステップオーバー」
（あるいは F10 キー）



char message[] = "hello";





メモリの中身は **68 65 6c 6c 6f 00** に**変化**

68→**h**
65→**e**
6c→**l**
6c→**l**
6f→**o**

ASCIIコード

00→**文字列の終わり**

- 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」

- プログラムを次のように書き換えて、同じことをもう1度行いなさい

```
7   int main()
8   {
9       char message[] = "aaa,bb cc";
10      printf("%s", message);
11      return 0;
12  }
```

メモリ 1

アドレス: 0x00F5FA28

```
0x00F5FA28  cc cc cc cc cc cc cc cc cc cc cc cc
0x00F5FA56  a0 00 0a 35 7c e9 46 10 a0 00 46
```

メモリ 1

アドレス: 0x00F5FA28

```
0x00F5FA28  61 61 61 2c 62 62 20 63 63 00 cc
0x00F5FA56  a0 00 0a 35 7c e9 46 10 a0 00 46
```