ca-12. スタック

（コンピュータ・アーキテクチャ演習）

URL: https://www.kkaneko.jp/cc/ca/index.html

金子邦彦

スタックのプッシュとポップ

- プッシュ (push)：
  スタックの一番上に追加
- ポップ (pop)：
  スタックの一番上から削除

スタックは,
複数のデータを格納できるデータ構造

演習

- Visual Studio を起動しなさい
- Visual Studio 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

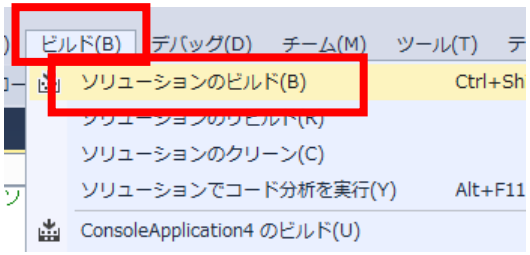
- Visual Studioのエディタを使って，ソースファイルを編集しなさい

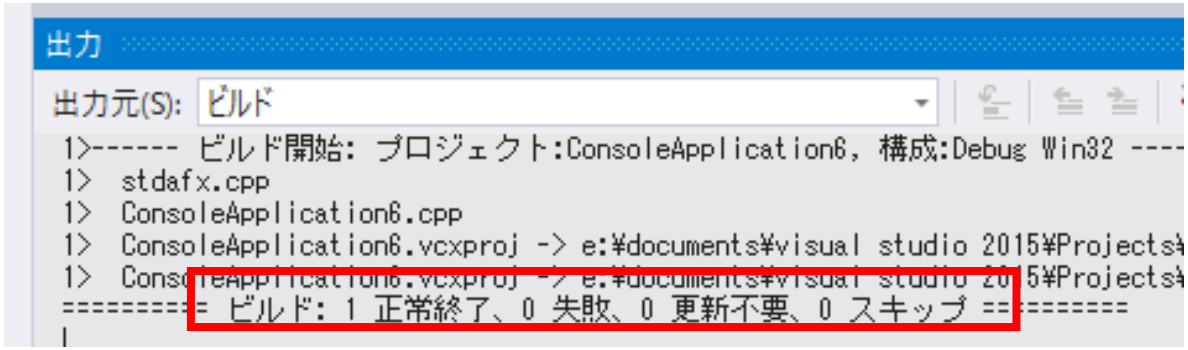
```
int main()
{
    _asm {
        push 10;
        push 5;
        push 20;
    }
    return 0;
}
```

追加

- 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で「push 10;」の行に，ブレークポイントを設定しなさい

```
int main()
{
    _asm {
        push 10;
        push 5;
        push 20;
    }
    return 0;
}
```

```
7
8
9
10
11
12
13
14
15
16
```

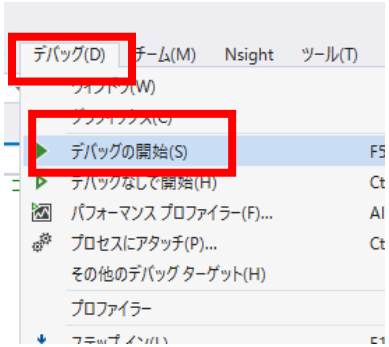
```
int main()
{
    _asm {
        push 10;
        push 5;
        push 20;
    }
    return 0;
}
```

① 「**push 10;**」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**」

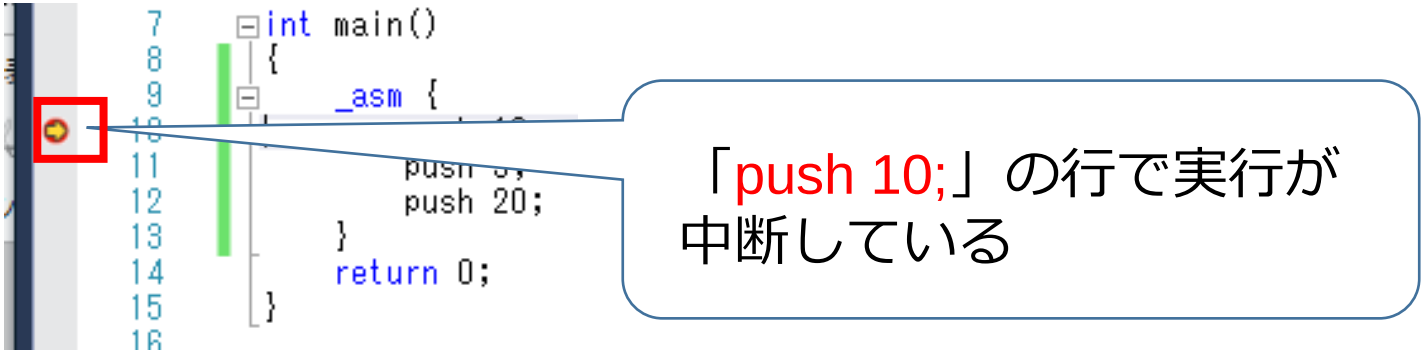
③ ブレークポイントが設定されるので確認．赤丸がブレークポイントの印

- Visual Studioで，デバッガーを起動しなさい.

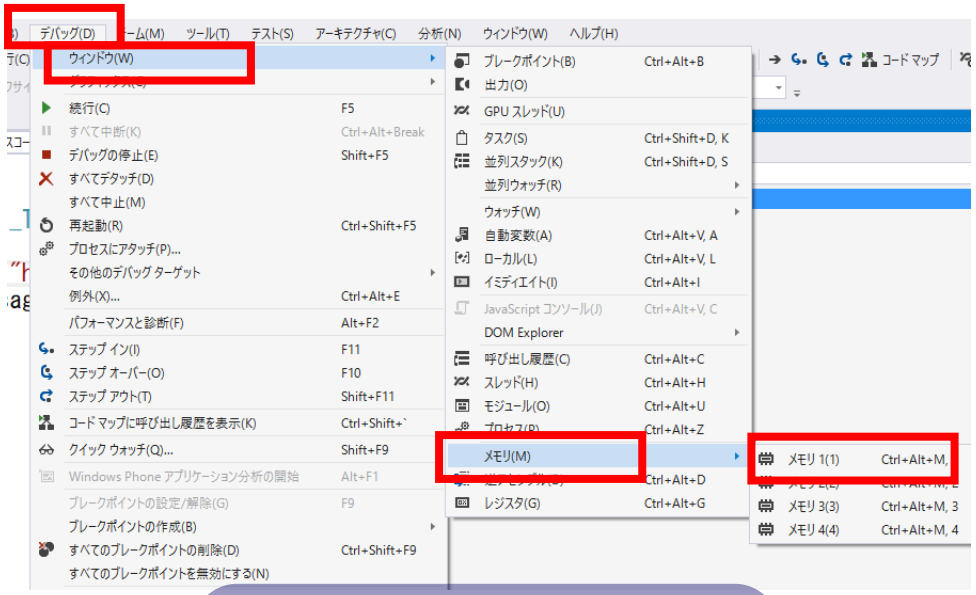


「デバッグ」
→ 「**デバッグ開始**」

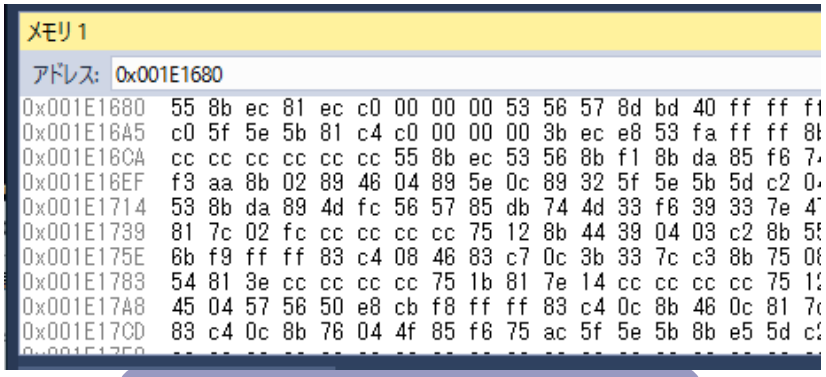
- 「push 10;」の行で，実行が中断することを確認しなさい
- あとで使うので，中断したままにしておくこと



- 「push 10;」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．

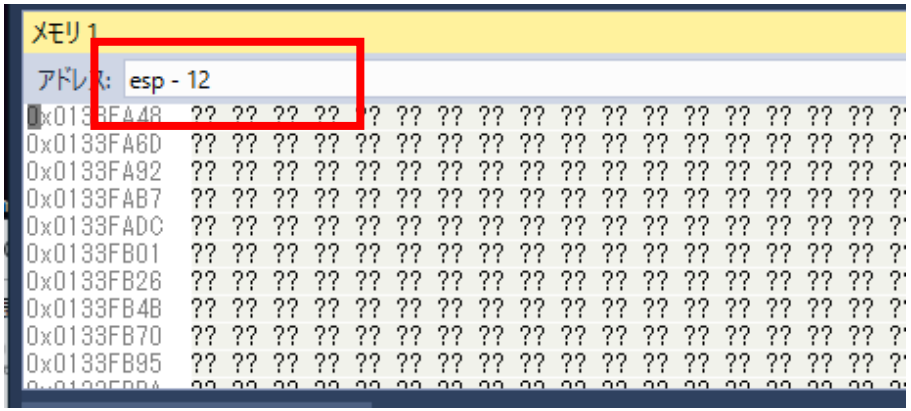


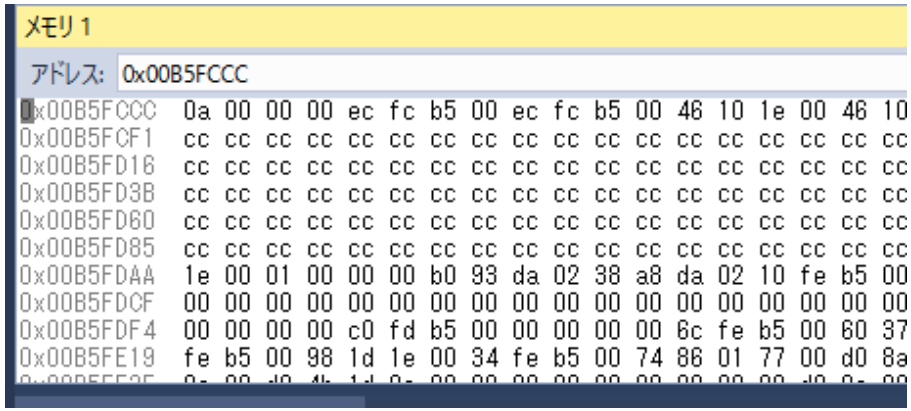
①「デバッグ」
→「ウインドウ」
→ 「**メモリ**」
→ 「**メモリ１（１）**」



②「メモリ１」の画面が表示される

- 「メモリ１」の画面の「アドレス」に「esp - 12」と入れて Enter キーを押しなさい

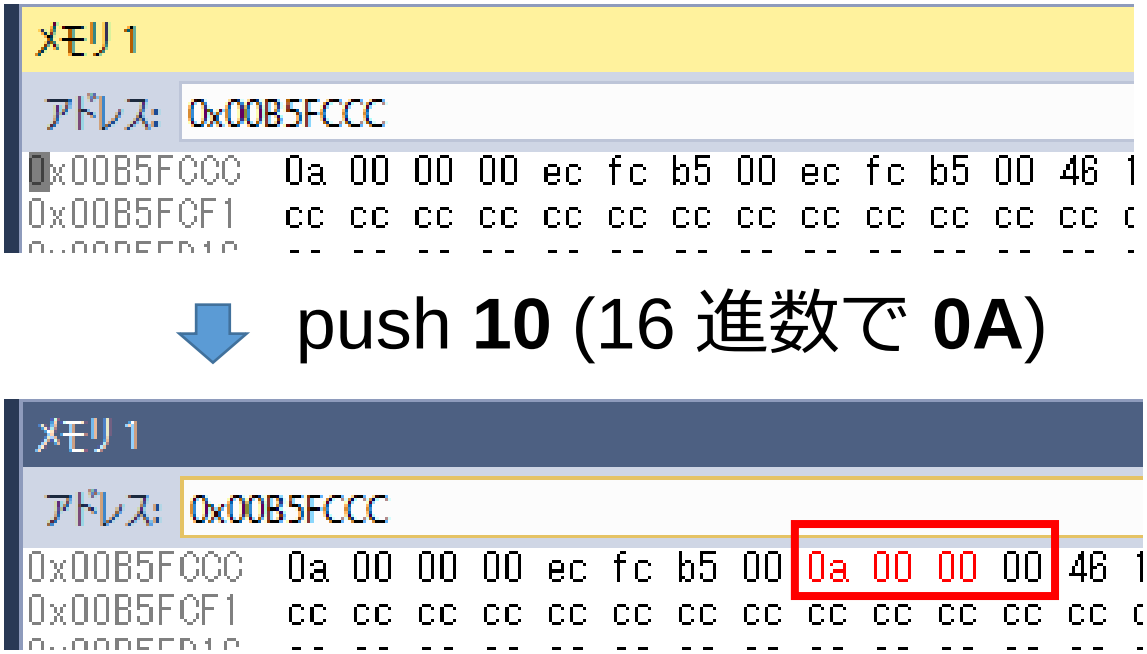




Enter キーを押すと
画面が変化するので確認する

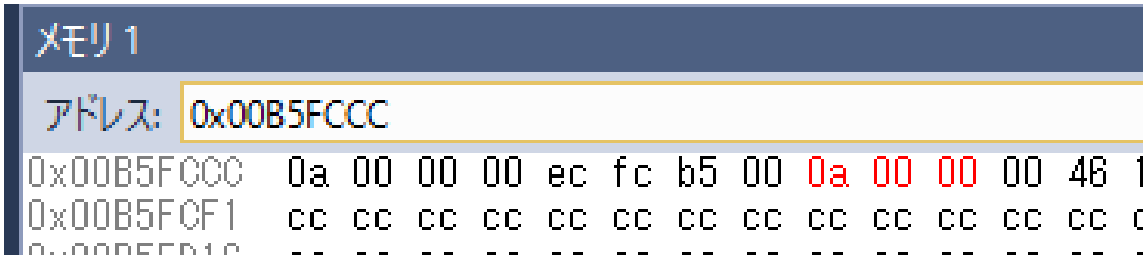
- ステップオーバーの操作を1回行い，変化を見なさい

「デバッグ」
→「ステップオーバー」
（あるいは F10 キー）

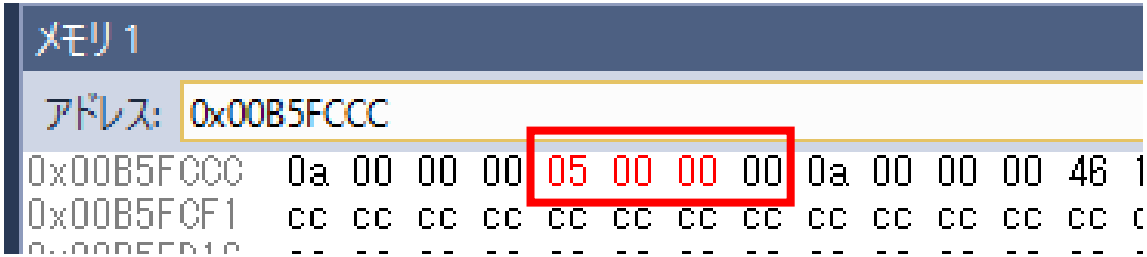


push **10** (16 進数で **0A**)

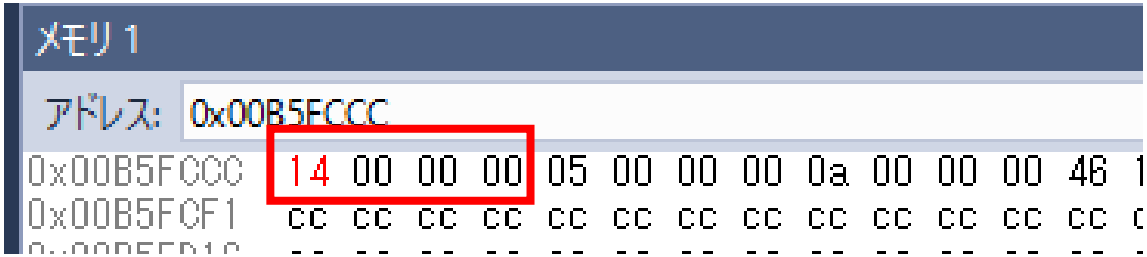
- ステップオーバーの操作を，さらに２回行い，変化を見なさい



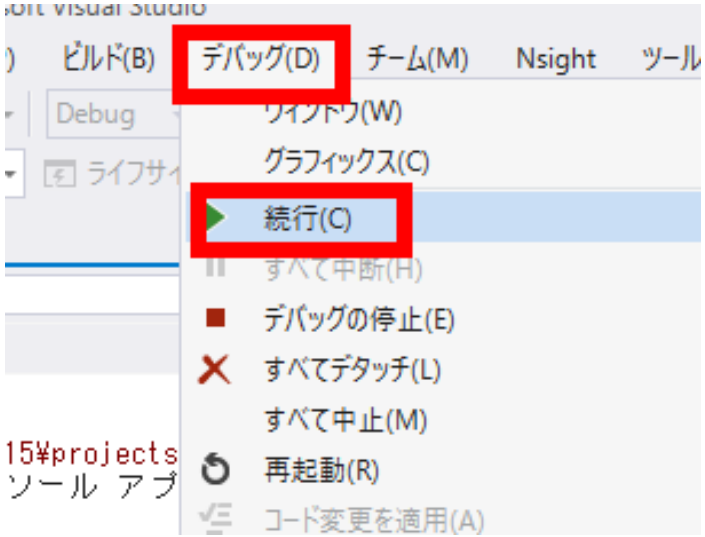
push **5** (16 進数で **5**)



push **20** (16 進数で **14**)

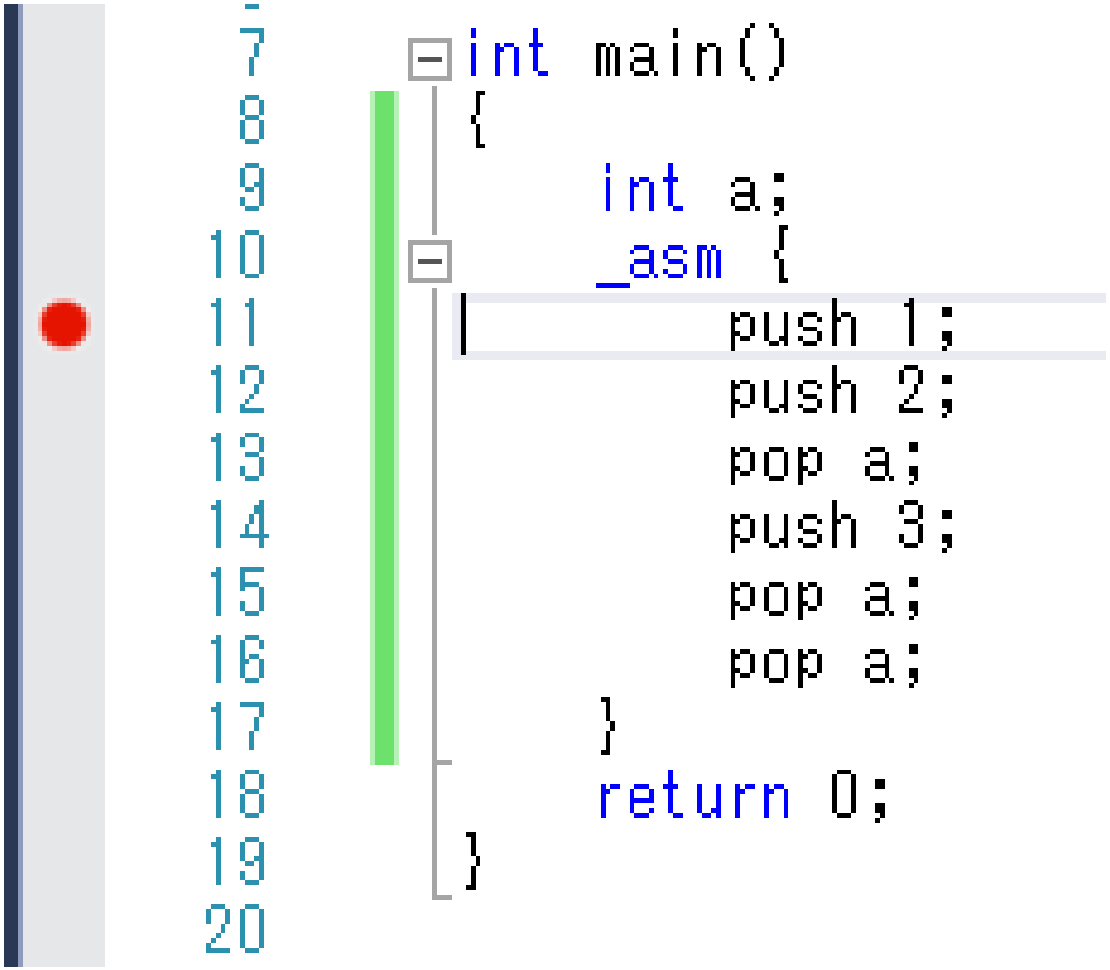


- 最後に，プログラム実行の再開の操作を行いなさい．これで，デバッガーが終了する．



「デバッグ」
→ 「**続行**」

- 次のように書き替えて，同じ手順を繰り返しなさい.



古いデータが消えずに残っている