ce-2. Microsoft Visual Studio C++ を使ってみよう

（C プログラミング応用）（全１４回）

URL: https://www.kkaneko.jp/pro/c/index.html

金子邦彦

CC BY NC SA

Microsoft Visual Studio Cを用いたC/Cプログラムの作成・ビルド・実行の一連の手順の習得。

【学習内容の構成】

1. **プロジェクトの新規作成**：Win32コンソールアプリケーションとしてプロジェクトを作成し、保存場所を設定
2. **C++ソースファイルの編集**：編集画面でプログラムコードを記述
3. **ビルドと実行**：ソリューションのビルドによるコンパイルと、デバッグなしで開始による実行
4. **他のソフトとの連携**：プログラムが出力したCSVファイルをMicrosoft Excelで読み込みグラフ作成

- 前提：基本的なPC操作
- 意義：プログラムによる計算の自動化とデータ活用の体験

例題１．プログラム実行の体験

- コンピュータを役に立つ道具として実感する
- 次ページのプログラムを使って，Microsoft Visual Studio C++でのC++ソースファイル編集，ビルド，テスト実行の一連の過程を体験する

例題１のプログラムの機能

- 計算の繰り返し
- キーボードからのデータ読み込み
- ファイルへの書き出し

```
#include "stdio.h"
#include <math.h>
#pragma warning(disable:4996)
int main()
{
    double x;
    double y;
    char buf[256];
    int i;
    double start_x;
    double step_x;
    FILE* fp;
    printf( "start_x =" );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf¥n", &start_x );
    printf( "step_x =" );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf¥n", &step_x );
    fp = fopen( "d:¥¥data.csv", "w" );
    for( i = 0; i < 20; i++ ) {
        x = start_x + ( i * step_x );
        y = sin( x );
        printf( "x= %f, y= %f¥n", x, y );
        fprintf( fp, "x=, %f, y=, %f¥n", x, y );
    }
    fprintf( stderr, "file d:¥¥data.csv created¥n" );
    fclose( fp );
    return 0;
}
```

データファイル名
d:¥¥data.csv
は適切に設定すること

キーボードからの
データ読み込みを
行っている部分

計算を行っている部分

ファイルへの書き出し
を行っている部分

```
#include "stdio.h"
#include <math.h>
#pragma warning(disable:4996)
int main()
{
    double x;
    double y;
    char buf[256];
    int i;
    double start_x;
    double step_x;
    FILE* fp;
    printf( "start_x =" );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf¥n", &start_x );
    printf( "step_x =" );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf¥n", &step_x );
    fp = fopen( "d:¥¥data.csv", "w" );
    for( i = 0; i < 20; i++ ) {
        x = start_x + ( i * step_x );
        y = sin( x );
        printf( "x= %f, y= %f¥n", x, y );
        fprintf( fp, "x=, %f, y=, %f¥n", x, y );
    }
    fprintf( stderr, "file d:¥¥data.csv created¥n" );
    fclose( fp );
    return 0;
}
```

Cプログラムはメインの関数から実行開始

変数 **x, y, buf, i, start_x, step_x, fp** をメモリエリア中に確保

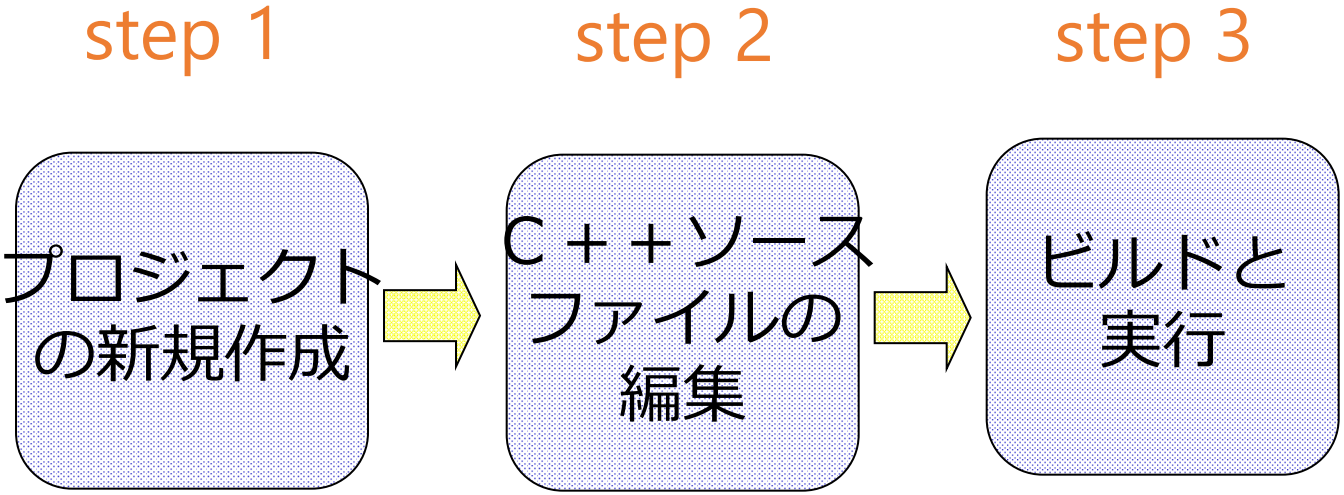
「**start_x=**」というメッセージを表示し，キーボードから数値を読み込む（読み込んだ値は変数 **start_x** に格納）

「**step_x=**」というメッセージを表示し，キーボードから数値を読み込む（読み込んだ値は **step_x** に格納）

20回の繰り返し (i = 0, 1, ... 19)

x の値から **sin(x)** を求め，**y** に書き込む

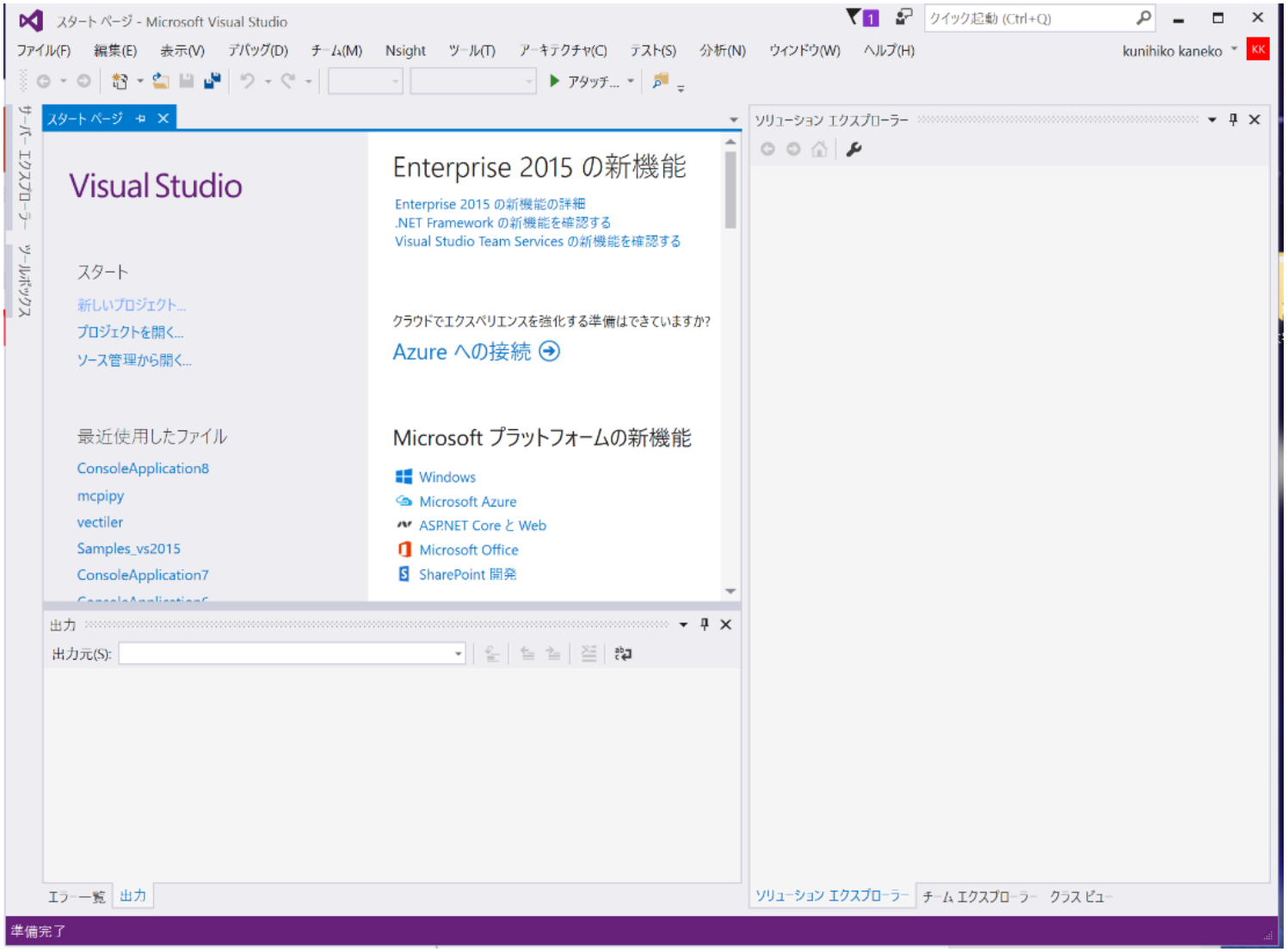
Microsoft Visual Studio C++での プログラム実行までの手順



Microsoft Visual Studio C++ の起動(1/2)

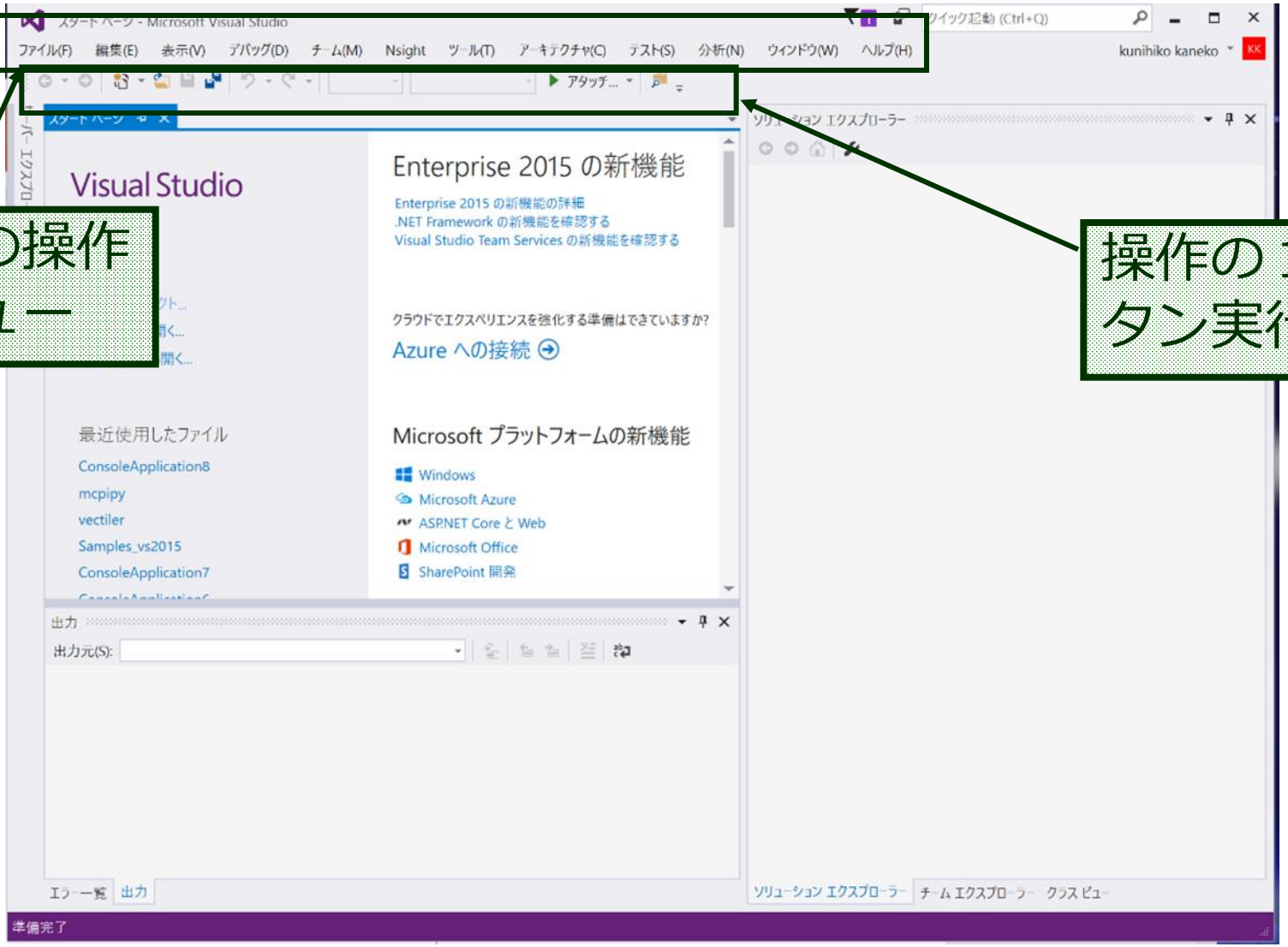
- 「スタート」
  - →「プログラム」
  - →「Visual Studio 2015」

Microsoft Visual Studio C++ の起動(2/2)

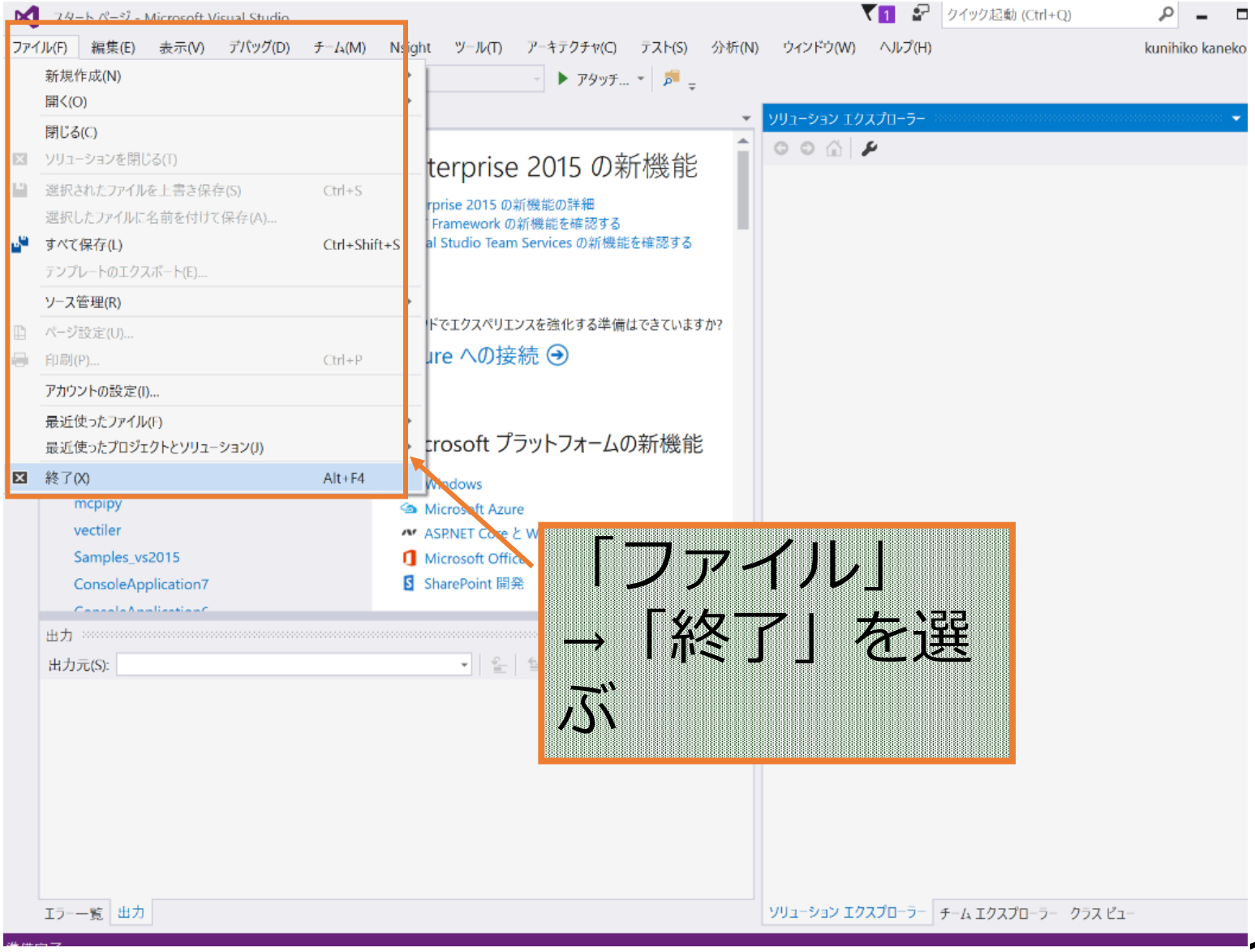


Microsoft Visual Studio 2015 の初回起動では，上のような画面が現

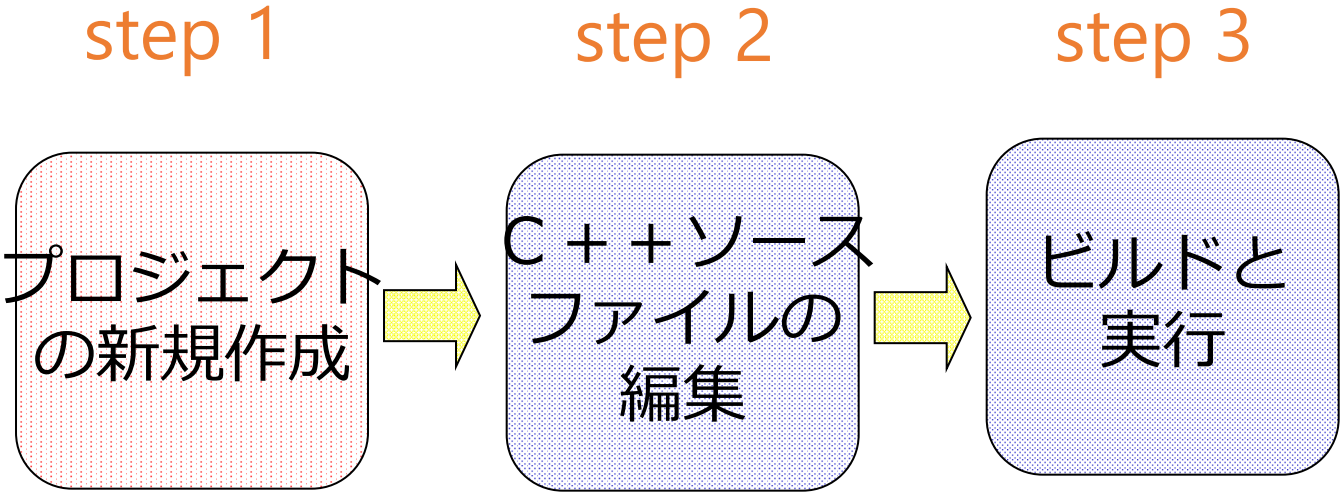
Microsoft Visual Studio C++ の起動(2/2)



Microsoft Visual Studio C++ の終了

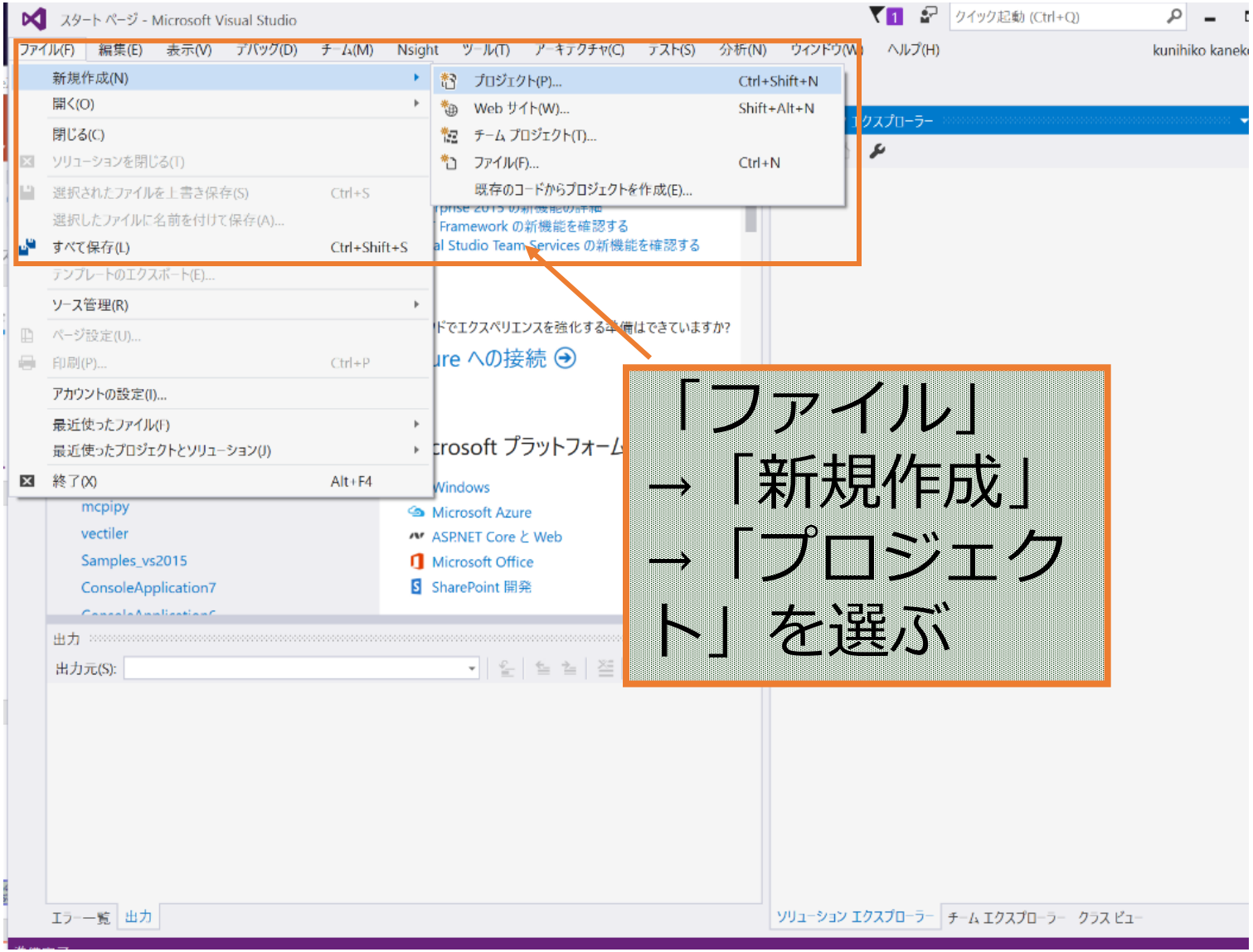


プロジェクトの新規作成

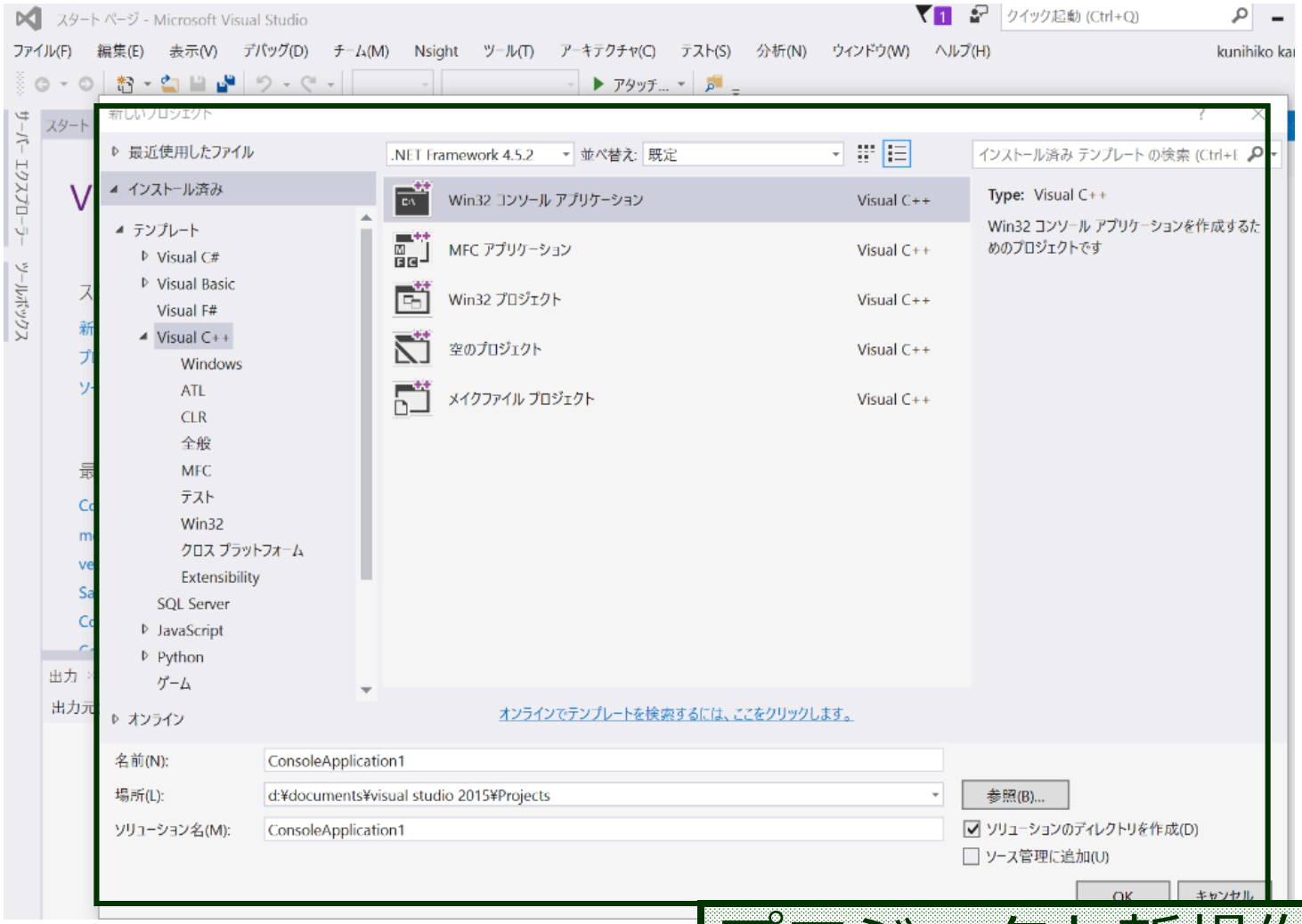


- 好きな「プロジェクト名」を入れる
- 好きな「場所」を入れる

プロジェクトの新規作成(1/8)

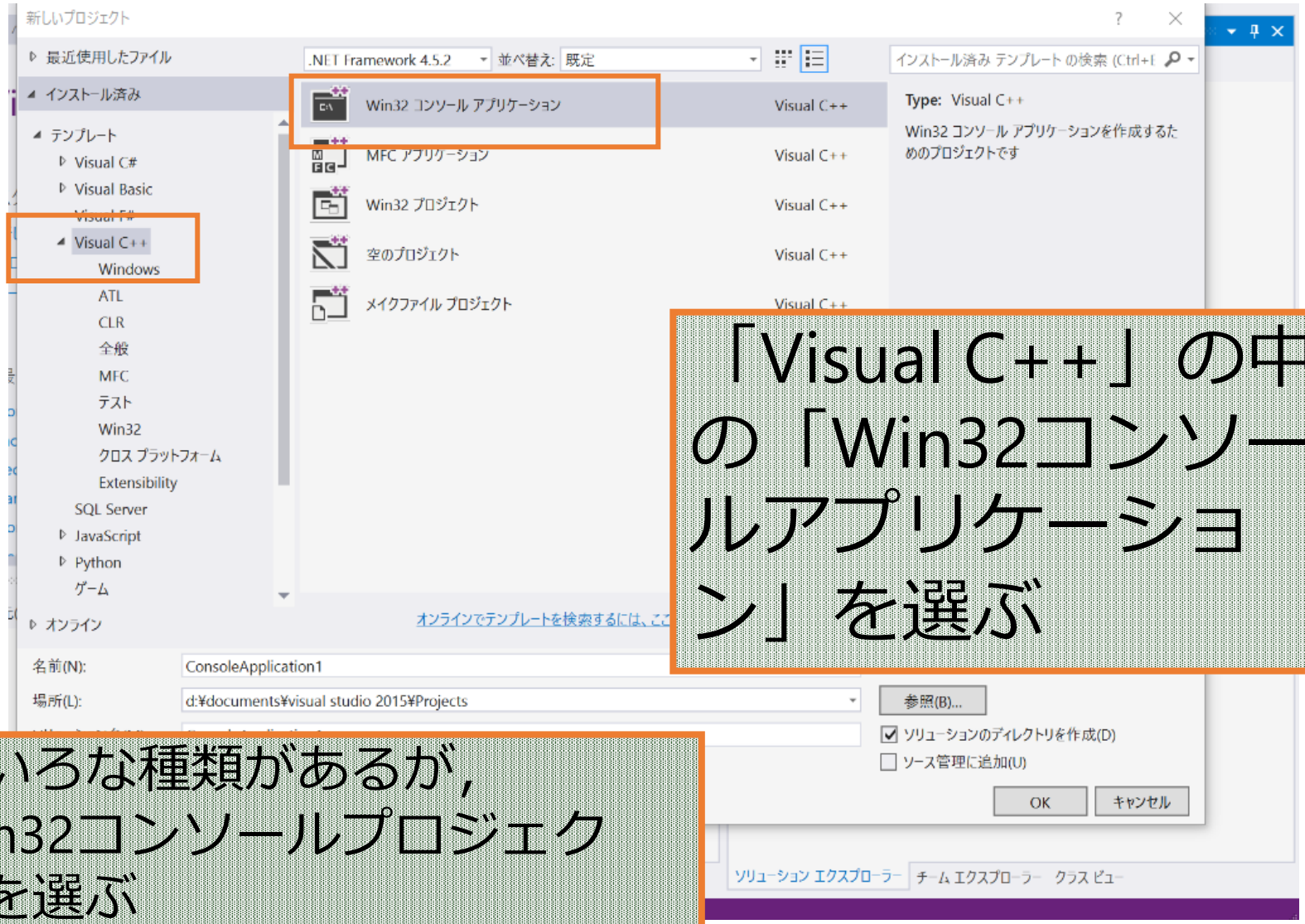


プロジェクトの新規作成(2/8)



プロジェクト新規作成のためのウインドウが現れる

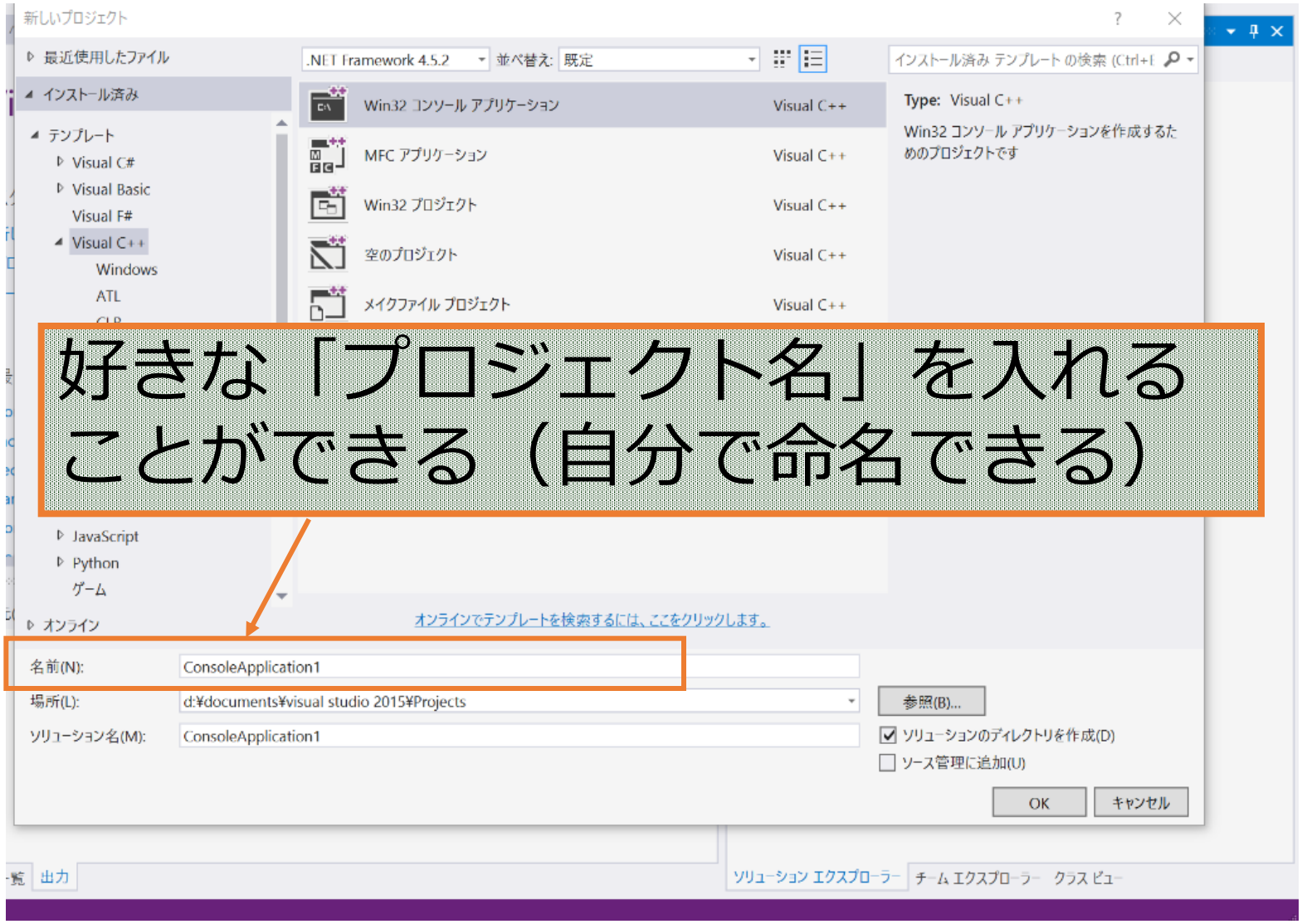
プロジェクトの新規作成(3/8)



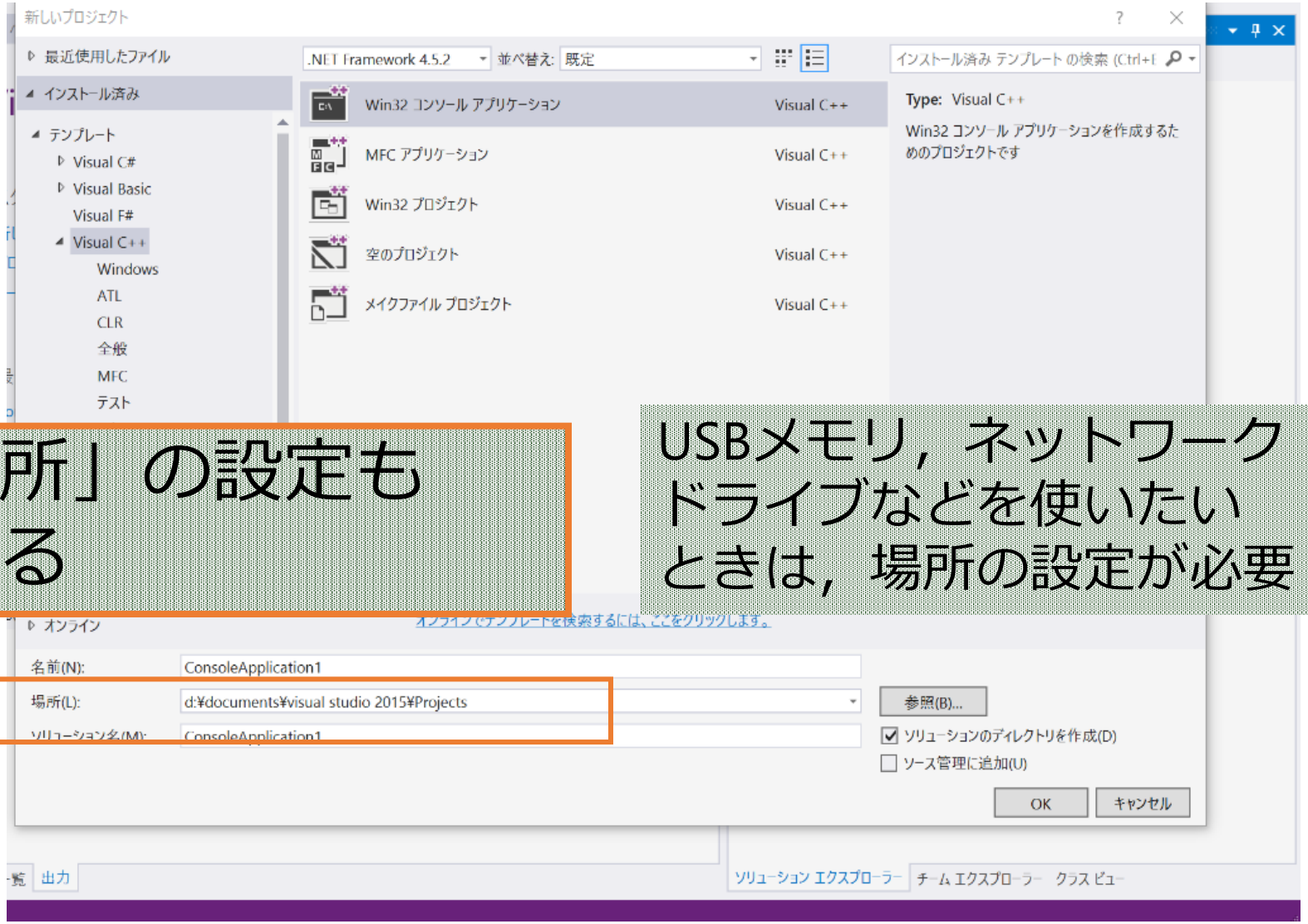
「Visual C++」の中の「Win32コンソールアプリケーション」を選ぶ

いろいろな種類があるが，「Win32コンソールプロジェクト」を選ぶ（それ以外は使用しない）

プロジェクトの新規作成(4/8)



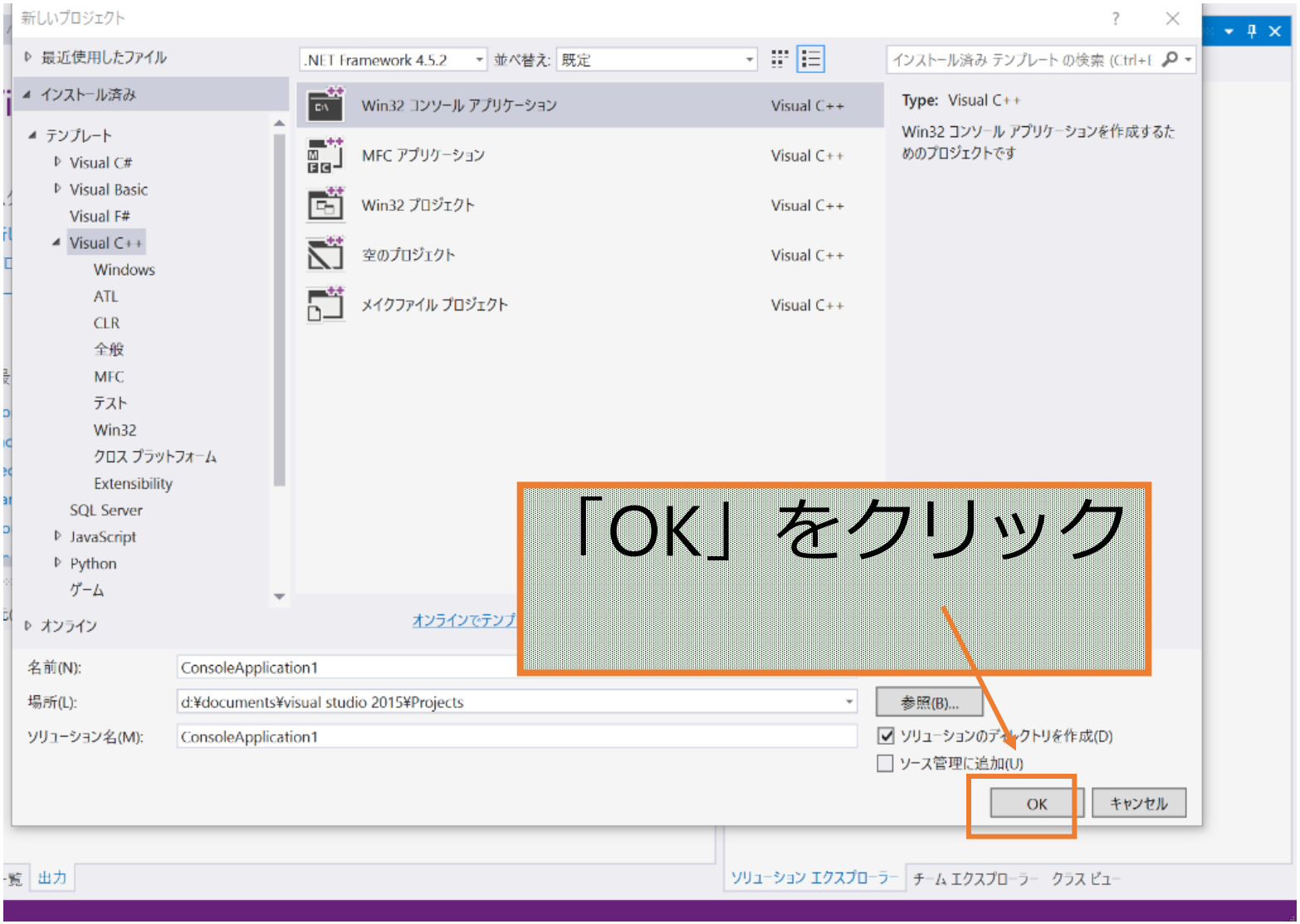
プロジェクトの新規作成(5/8)



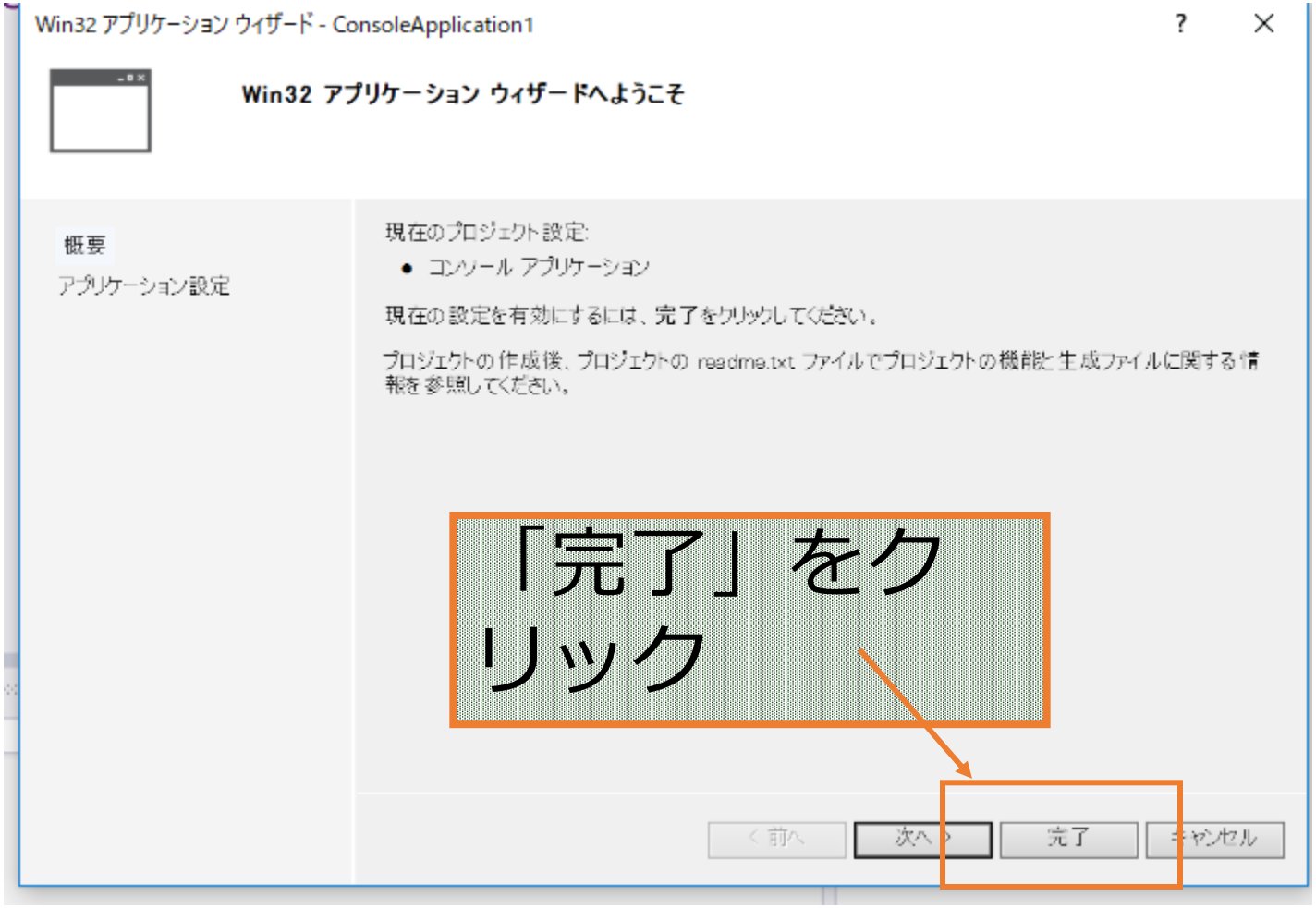
「場所」の設定もできる

USBメモリ，ネットワークドライブなどを使いたいときは，場所の設定が必要

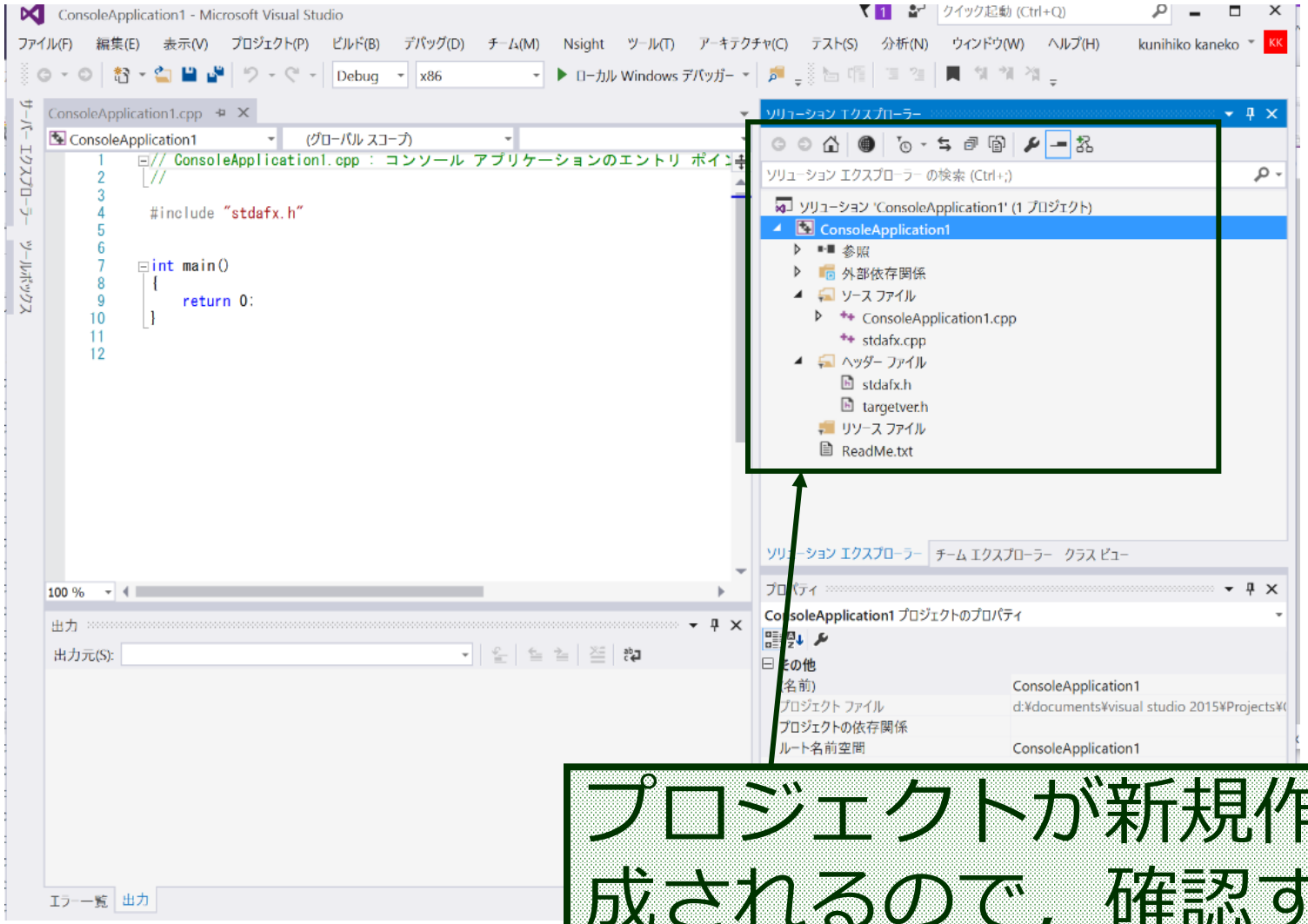
プロジェクトの新規作成(6/8)



プロジェクトの新規作成(7/8)



プロジェクトの新規作成



プロジェクトが新規作成されるので，確認する

プロジェクトの新規作成

① 「新規作成」→「プロジェクト」を選ぶ

② 新規作成のためのウインドウが現れる

③ プロジェクトの選択

「Win32 コンソールプロジェクト」を選ぶ

④ プロジェクト名の入力

好きな「プロジェクト名」を入れることができる

⑤ 場所の入力

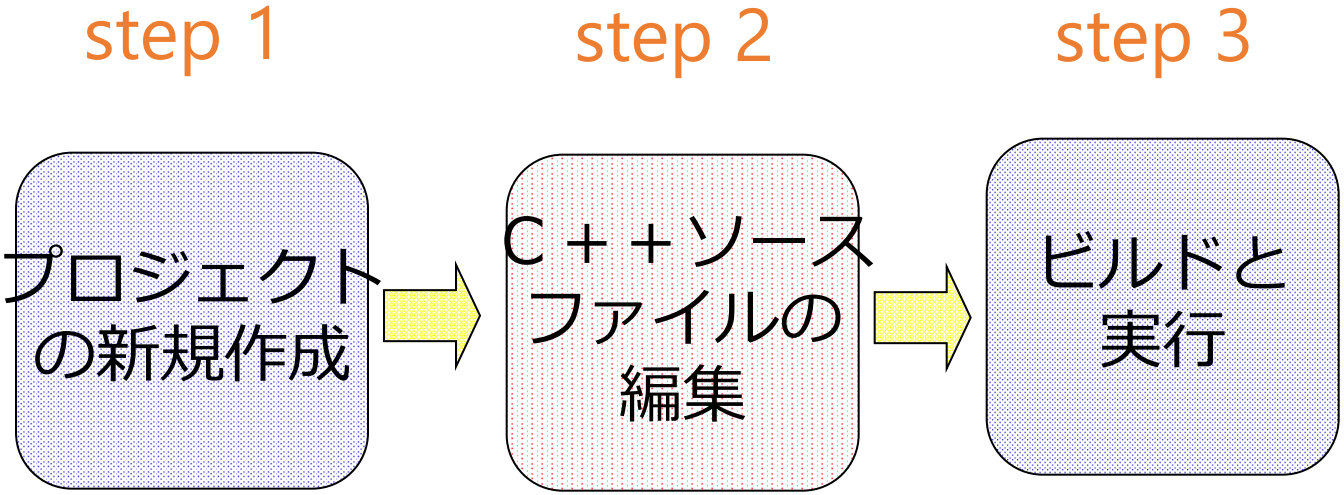
好きな「場所」を入れるできる

⑥ Win32アプリケーションウイザード

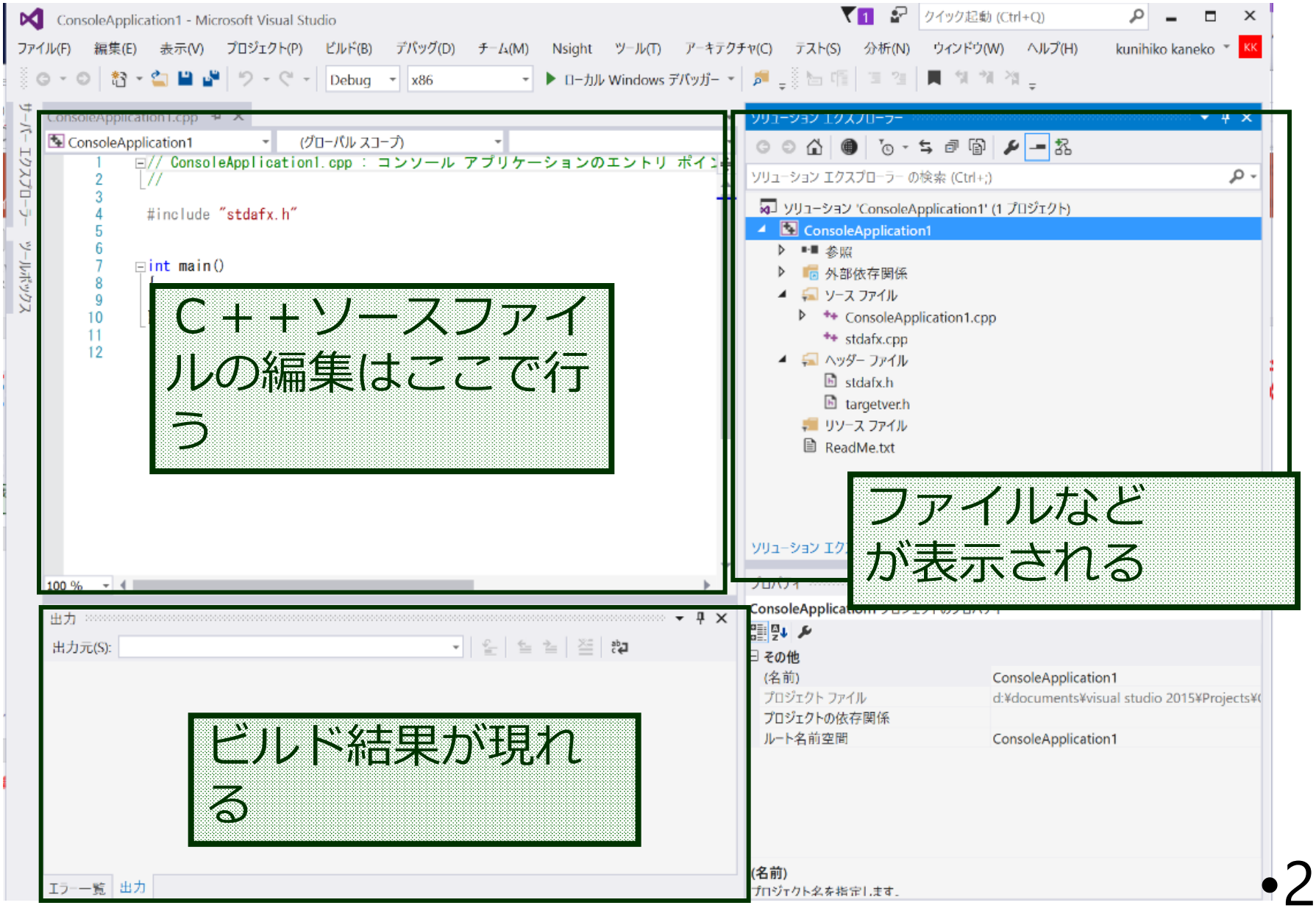
「完了」を選ぶ

⑦ 先ほど入力した「プロジェクト名」が現れるので，確認する

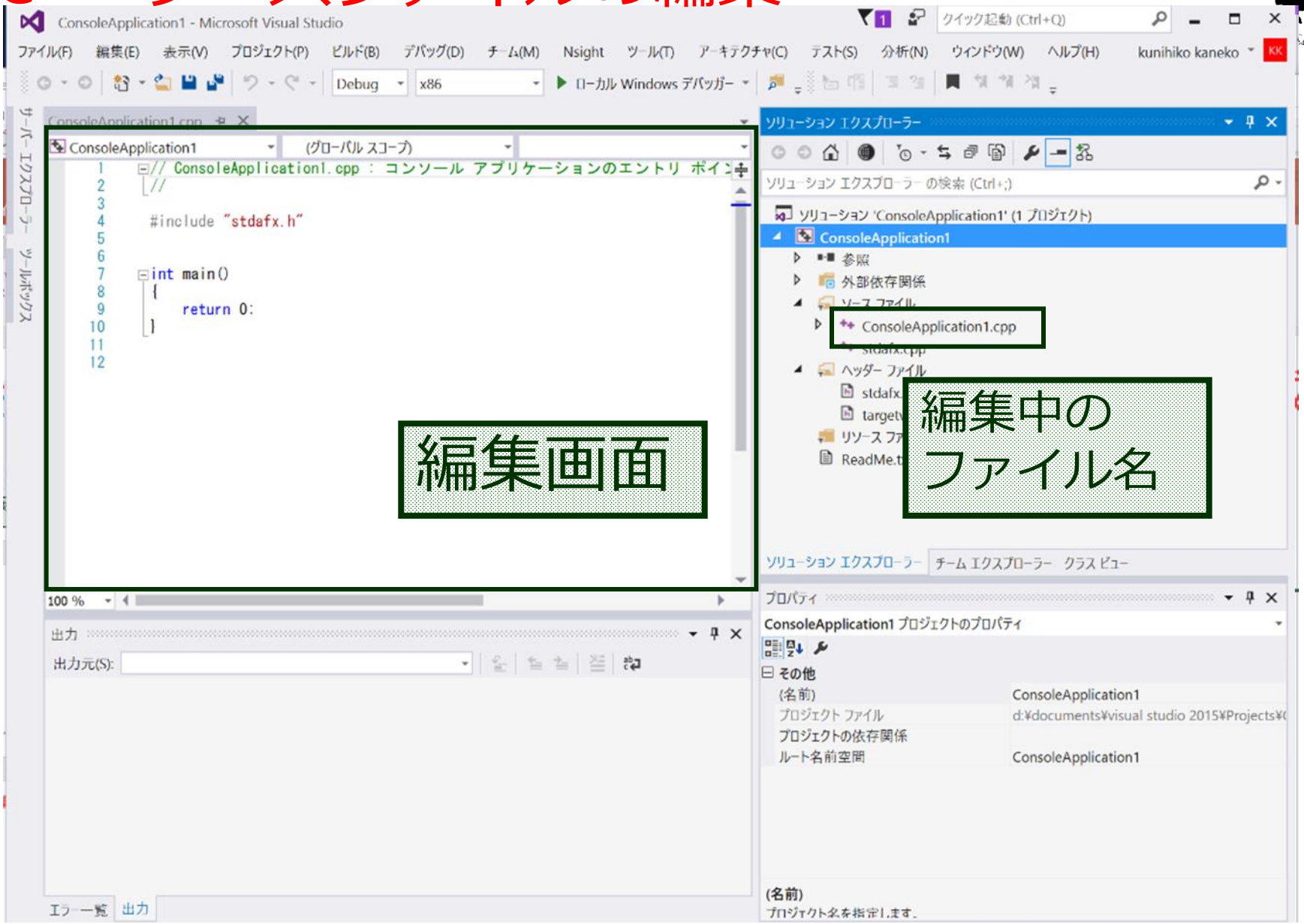
C＋＋ソースファイルの編集



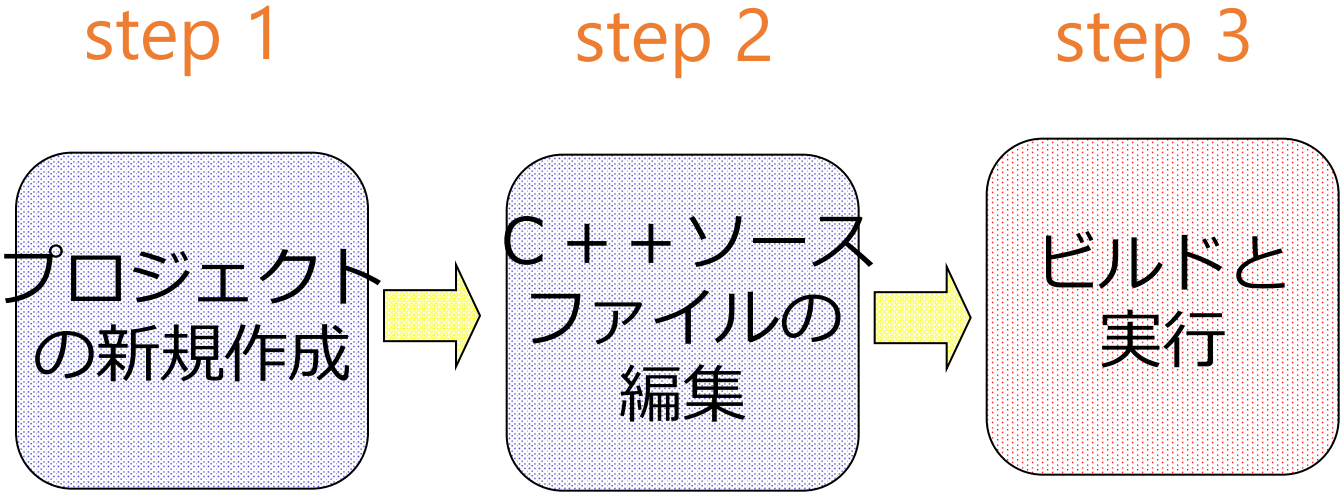
Microsoft Visual Studio C++ の画面構成



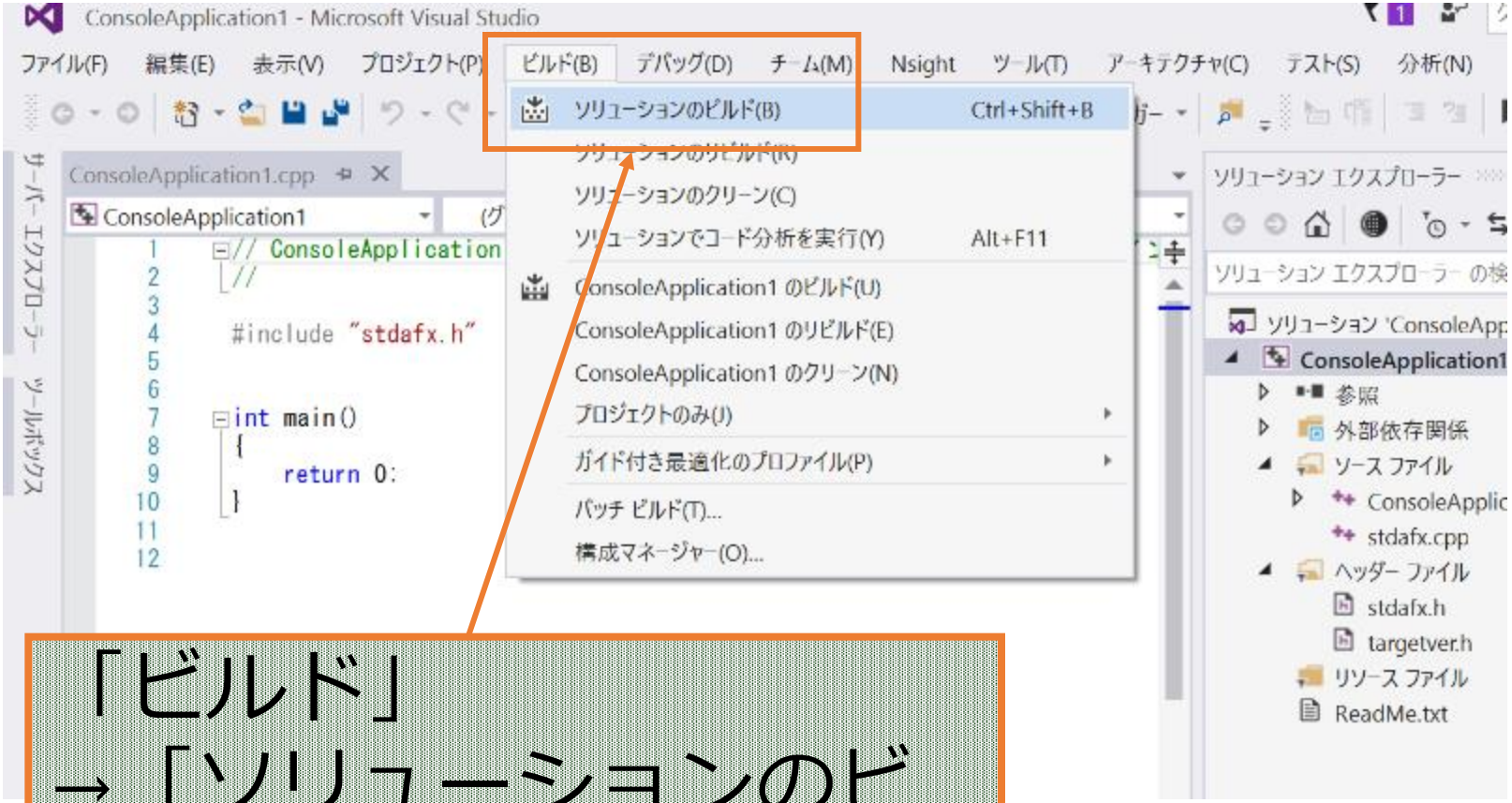
C++ ソースファイルの編集



Microsoft Visual Studio C++での プログラム実行までの手順

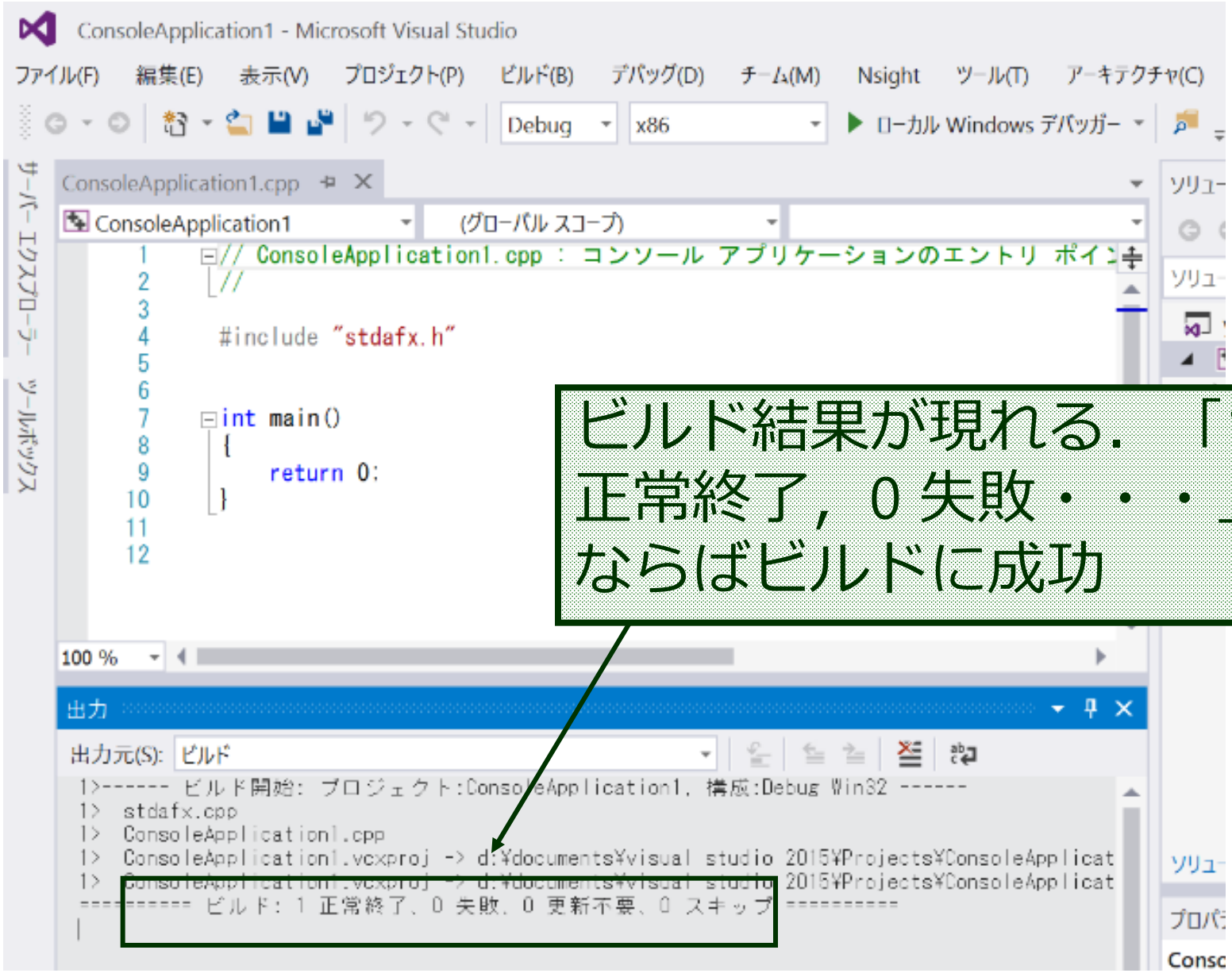


ビルドと実行(1/6)

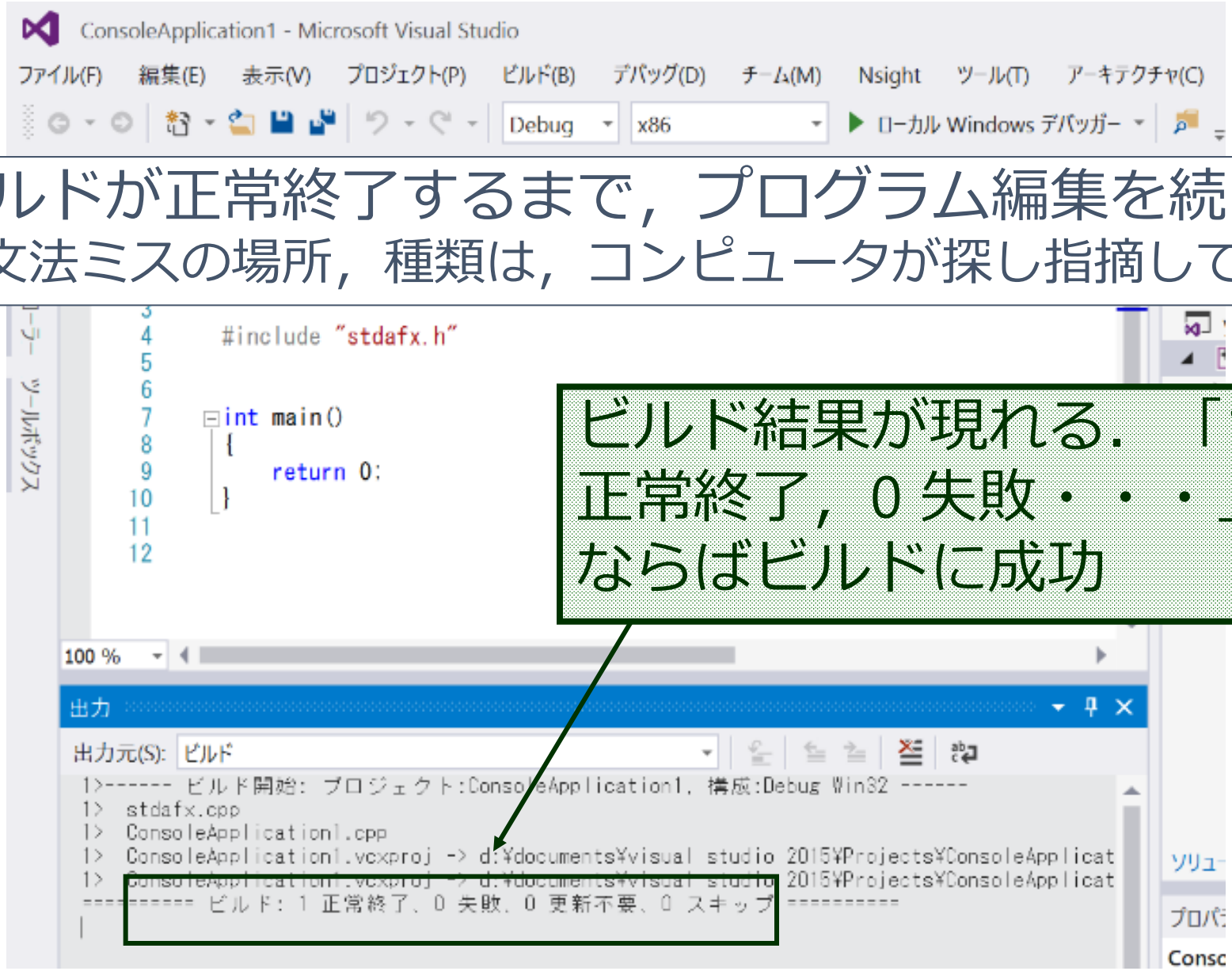


「ビルド」
→「ソリューションのビルド」を選ぶ

ビルドと実行(2/6)


ConsoleApplication1 - Microsoft Visual Studio
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) Nsight ツール(T) アーキテクチャ(C)
Debug
x86
ローカル Windows デバッガー
ConsoleApplication1.cpp
ConsoleApplication1
(グローバル スコープ)
// ConsoleApplication1.cpp : コンソール アプリケーションのエントリ ポイン
//
#include "stdafx.h"
int main()
{
return 0;
}
100 %
出力
出力元(S): ビルド
1>------ ビルド開始: プロジェクト:ConsoleApplication1, 構成:Debug Win32 ------
1> stdafx.cpp
1> ConsoleApplication1.cpp
1> ConsoleApplication1.vcxproj -> d:¥documents¥visual studio 2015¥Projects¥ConsoleApplicat
1> ConsoleApplication1.vcxproj -> d:¥documents¥visual studio 2015¥Projects¥ConsoleApplicat
========== ビルド: 1 正常終了、0 失敗、0 更新不要、0 スキップ ==========
ビルド結果が現れる．「1正常終了，0失敗・・・」ならばビルドに成功

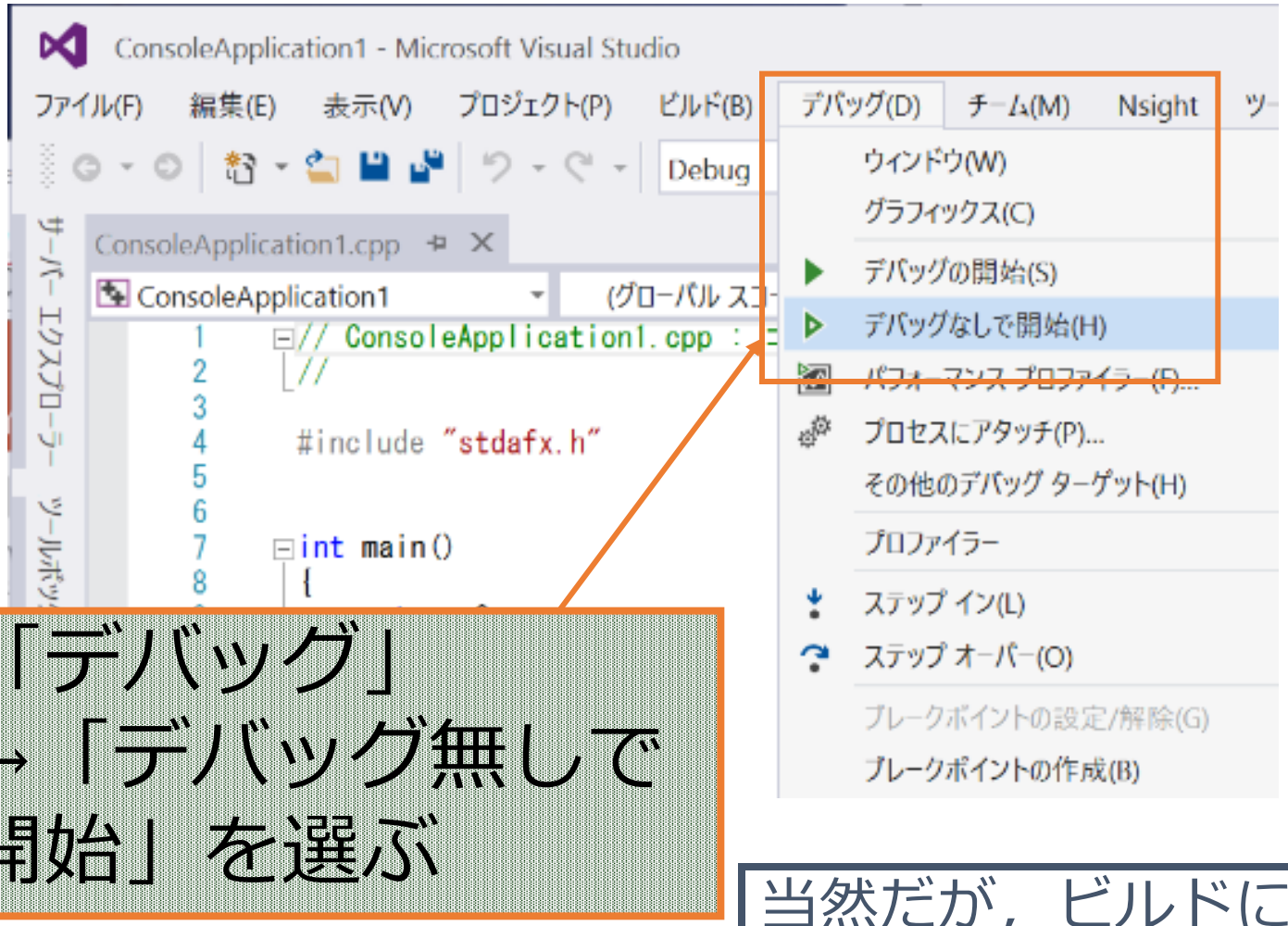
ビルドと実行(2/6)



ビルドが正常終了するまで，プログラム編集を続ける
・文法ミスの場所，種類は，コンピュータが探し指摘してくれる

ビルド結果が現れる．「1 正常終了，0 失敗・・・」ならばビルドに成功

ビルドと実行(3/6)



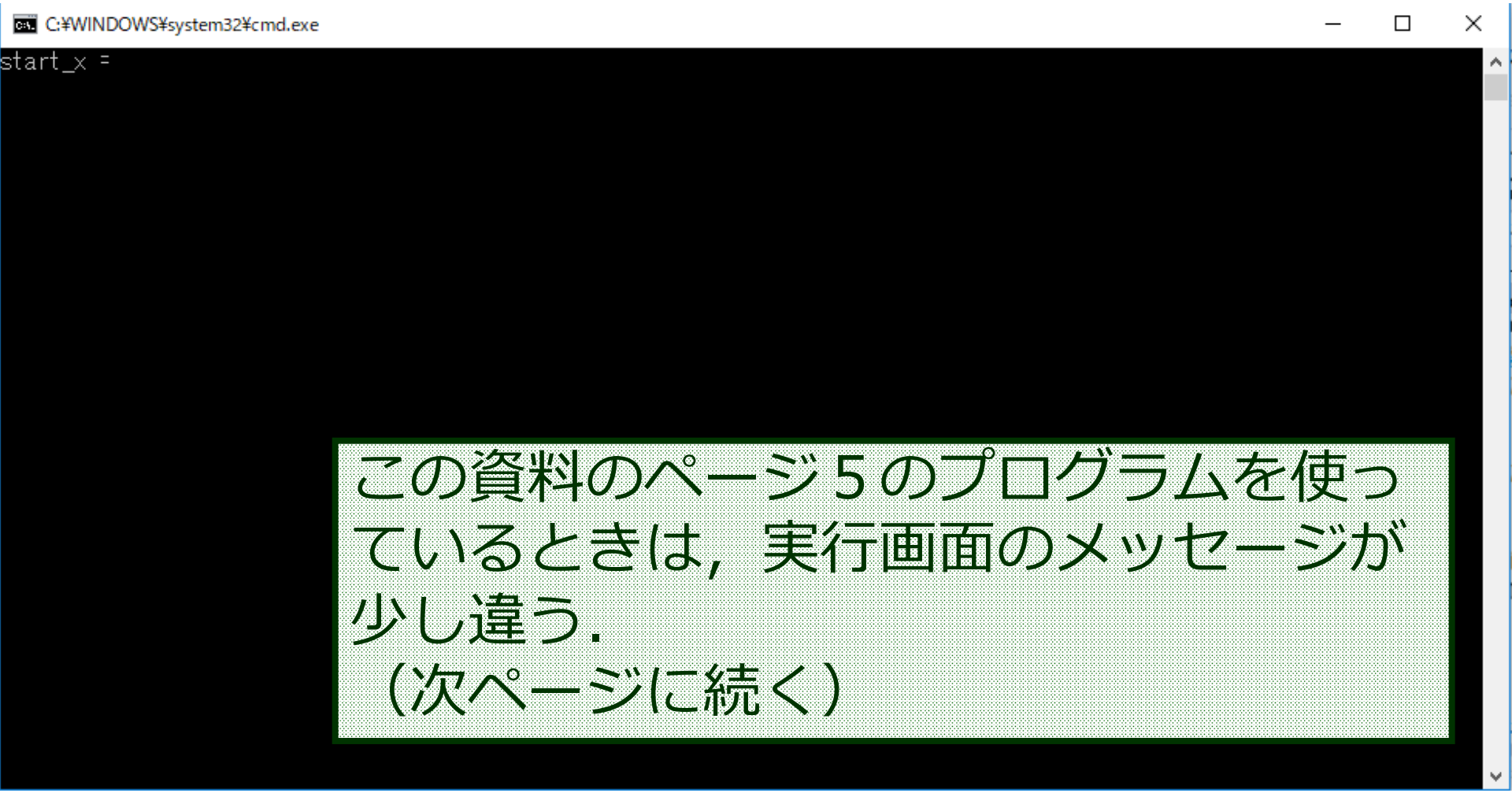
「デバッグ」
→「デバッグ無しで開始」を選ぶ

当然だが，ビルドに成功していないと，この操作は意味が無い

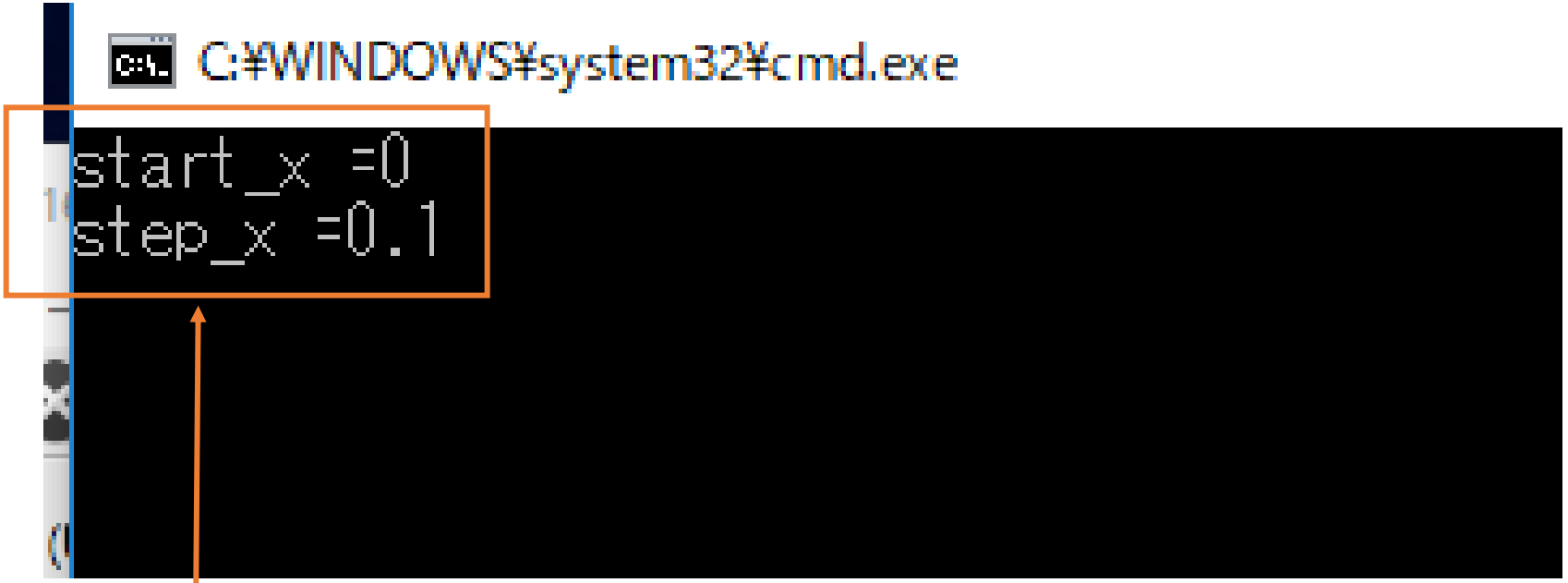
ビルドと実行(4/6)

実行画面が現れる

ビルドと実行について

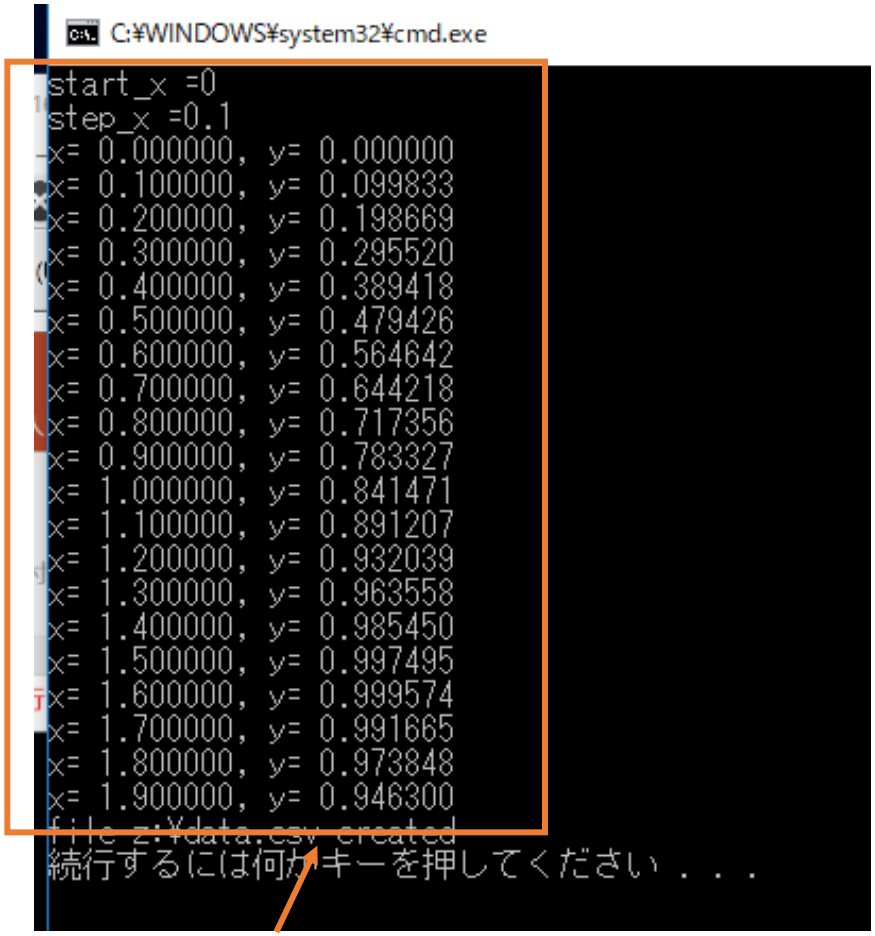

C:¥WINDOWS¥system32¥cmd.exe
start_x =
この資料のページ５のプログラムを使っているときは，実行画面のメッセージが少し違う．
（次ページに続く）

ビルドと実行について



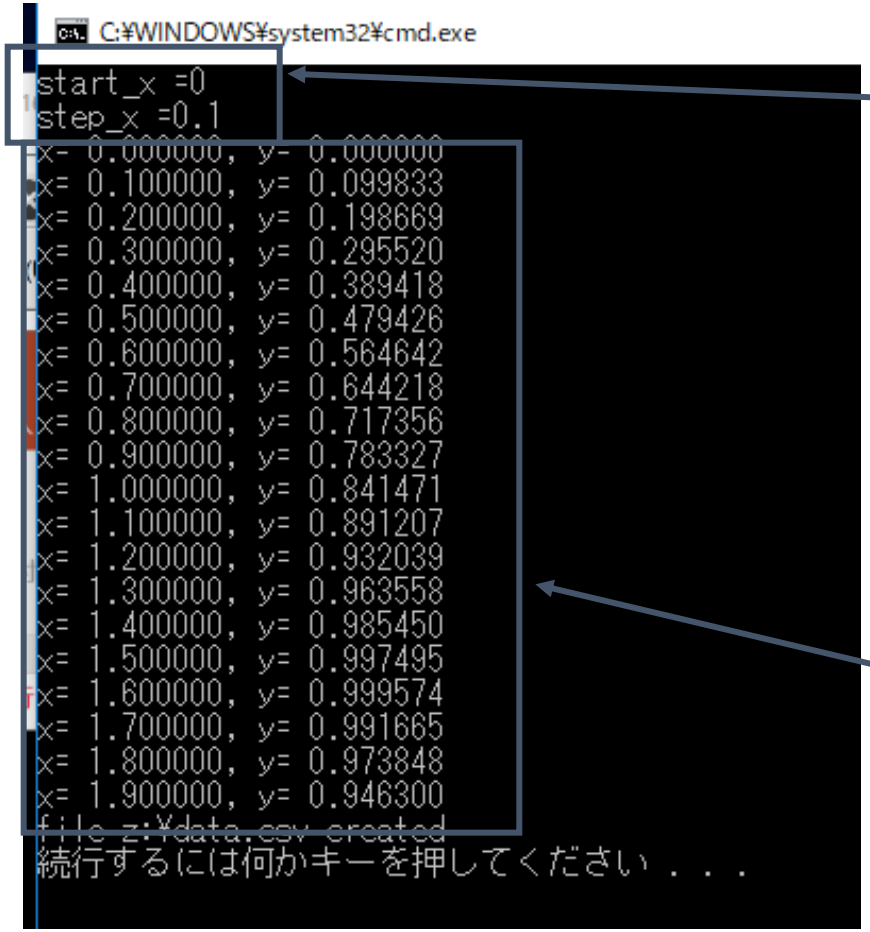
数値を入れる
（プログラムに数値データを与える）
start_x = 0
step_x = 0.1

ビルドと実行について



２つの数値を入れると，
計算結果の結果が得られる．

プログラム実行結果



キーボードから，データ「０」，「０．１」を読み込んでいる

計算を２０回繰り返して，計算結果を表示している

ビルドと実行

① 「ビルド」→「ソリューションのビルド」を選ぶ
　　ビルド結果が現れる
② 「デバッグ」→「デバッグ無しで開始」を選ぶ

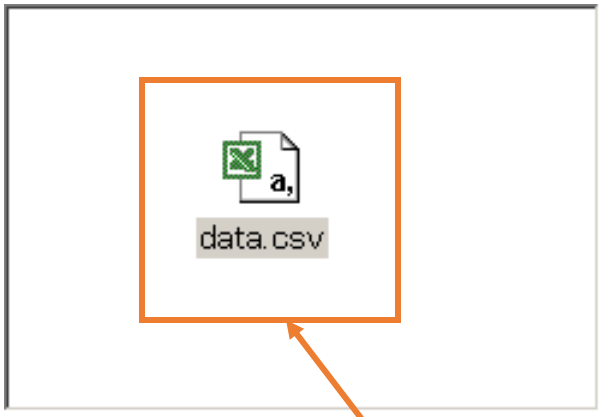
③ 実行画面が出る

例題２．他のソフトとの連携

- 例題１のプログラムによって書き出されたファイルを使い，Microsoft Excel を使って折れ線グラフを作成する

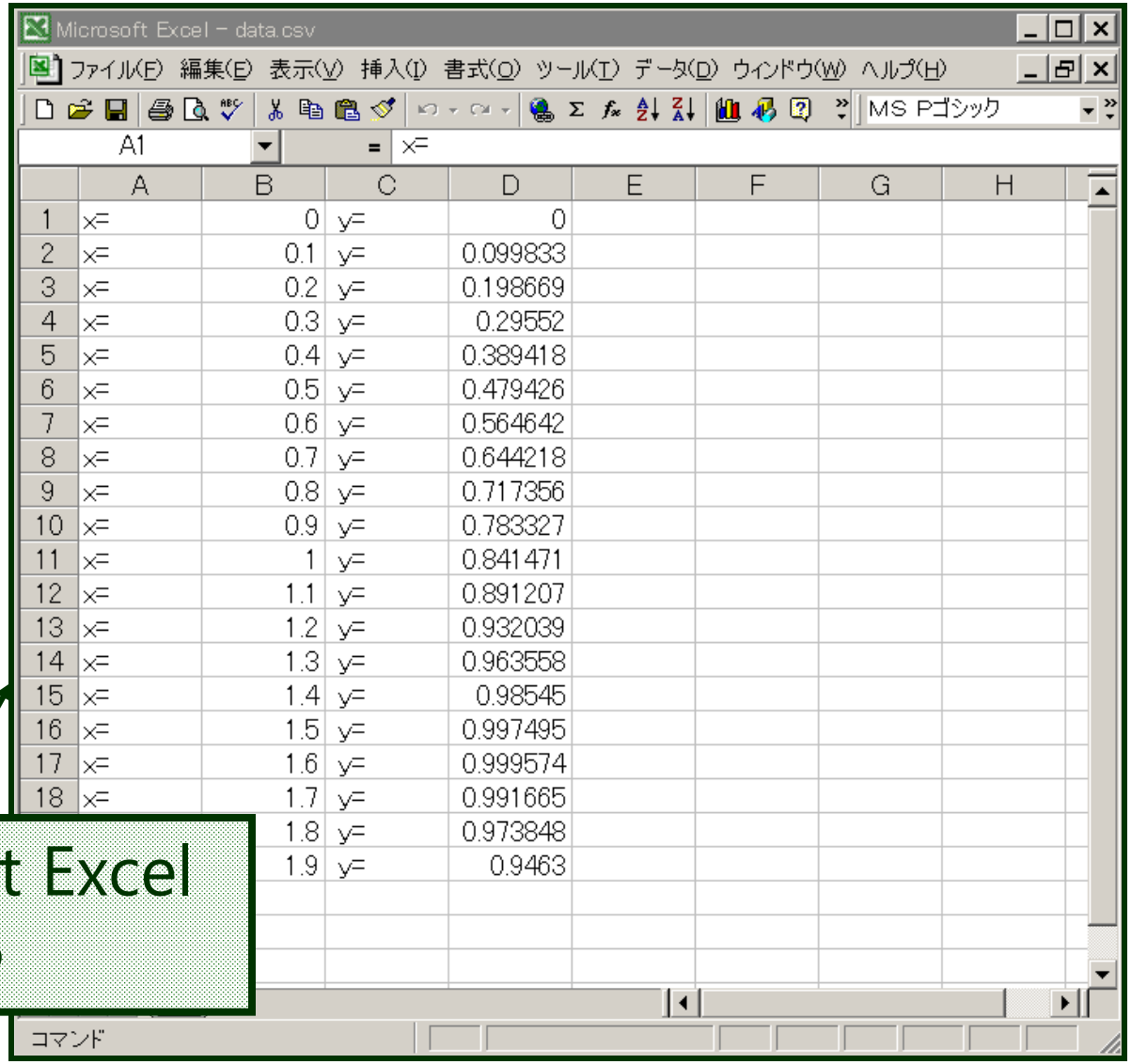
  ファイル名：　d:¥data.csv

Microsoft Excel でグラフ作成 (1/8)



z ドライブの
data.csv をダブルクリックする

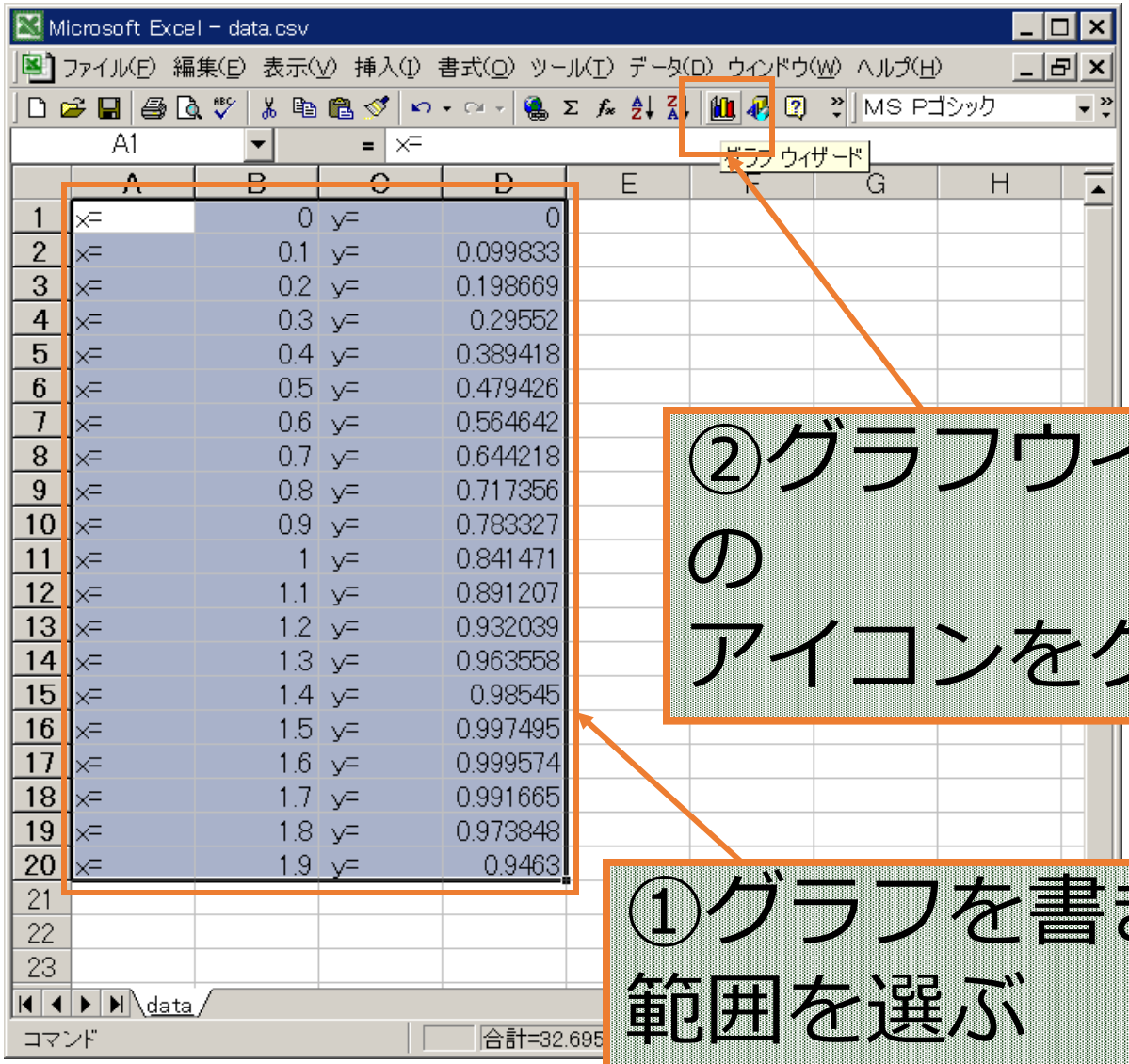
Microsoft Excel でグラフ作成 (2/8)



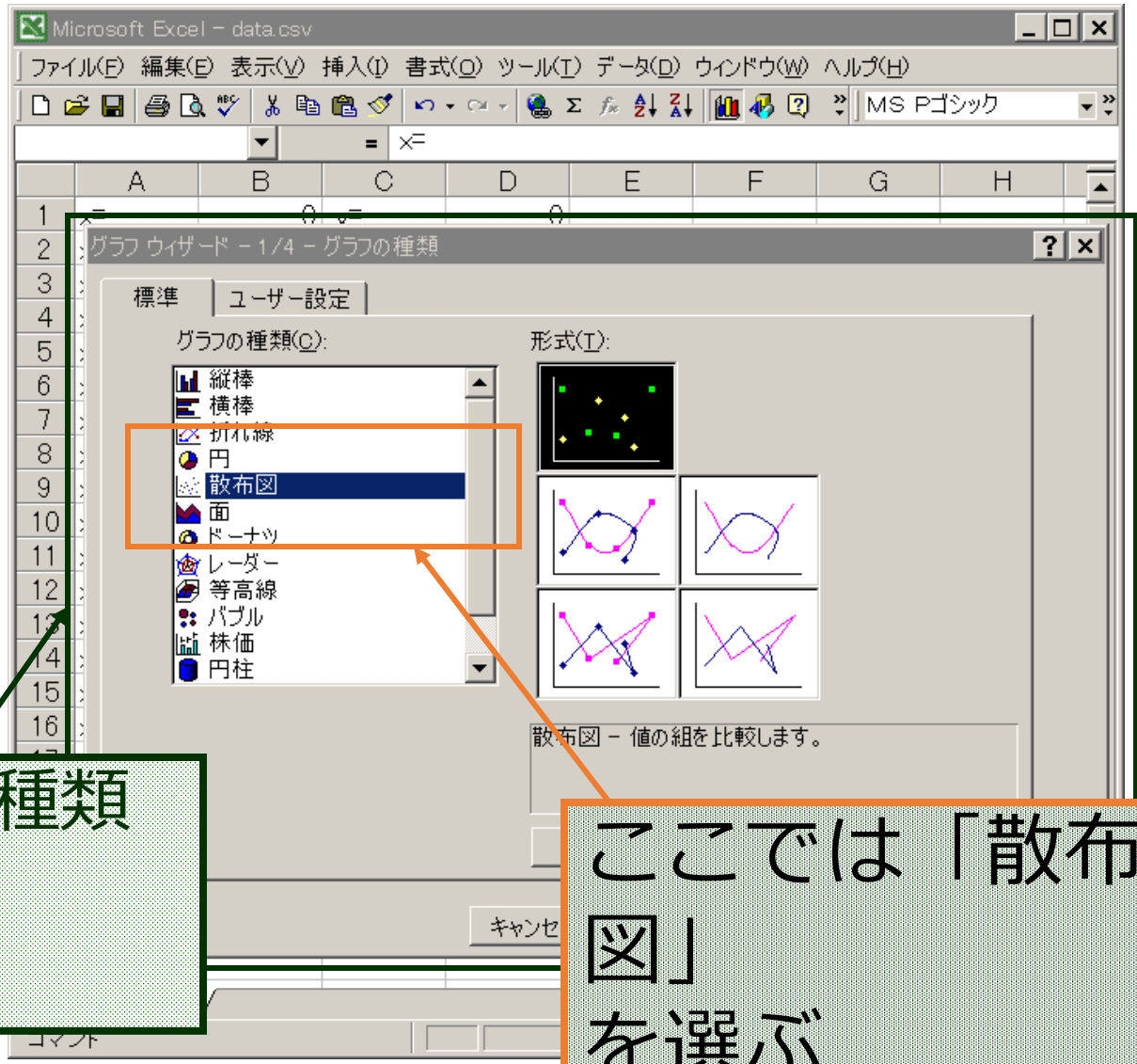
	A	B	C	D
1	x=	0	y=	0
2	x=	0.1	y=	0.099833
3	x=	0.2	y=	0.198669
4	x=	0.3	y=	0.29552
5	x=	0.4	y=	0.389418
6	x=	0.5	y=	0.479426
7	x=	0.6	y=	0.564642
8	x=	0.7	y=	0.644218
9	x=	0.8	y=	0.717356
10	x=	0.9	y=	0.783327
11	x=	1	y=	0.841471
12	x=	1.1	y=	0.891207
13	x=	1.2	y=	0.932039
14	x=	1.3	y=	0.963558
15	x=	1.4	y=	0.98545
16	x=	1.5	y=	0.997495
17	x=	1.6	y=	0.999574
18	x=	1.7	y=	0.991665
		1.8	y=	0.973848
		1.9	y=	0.9463

Microsoft Excel が現れる

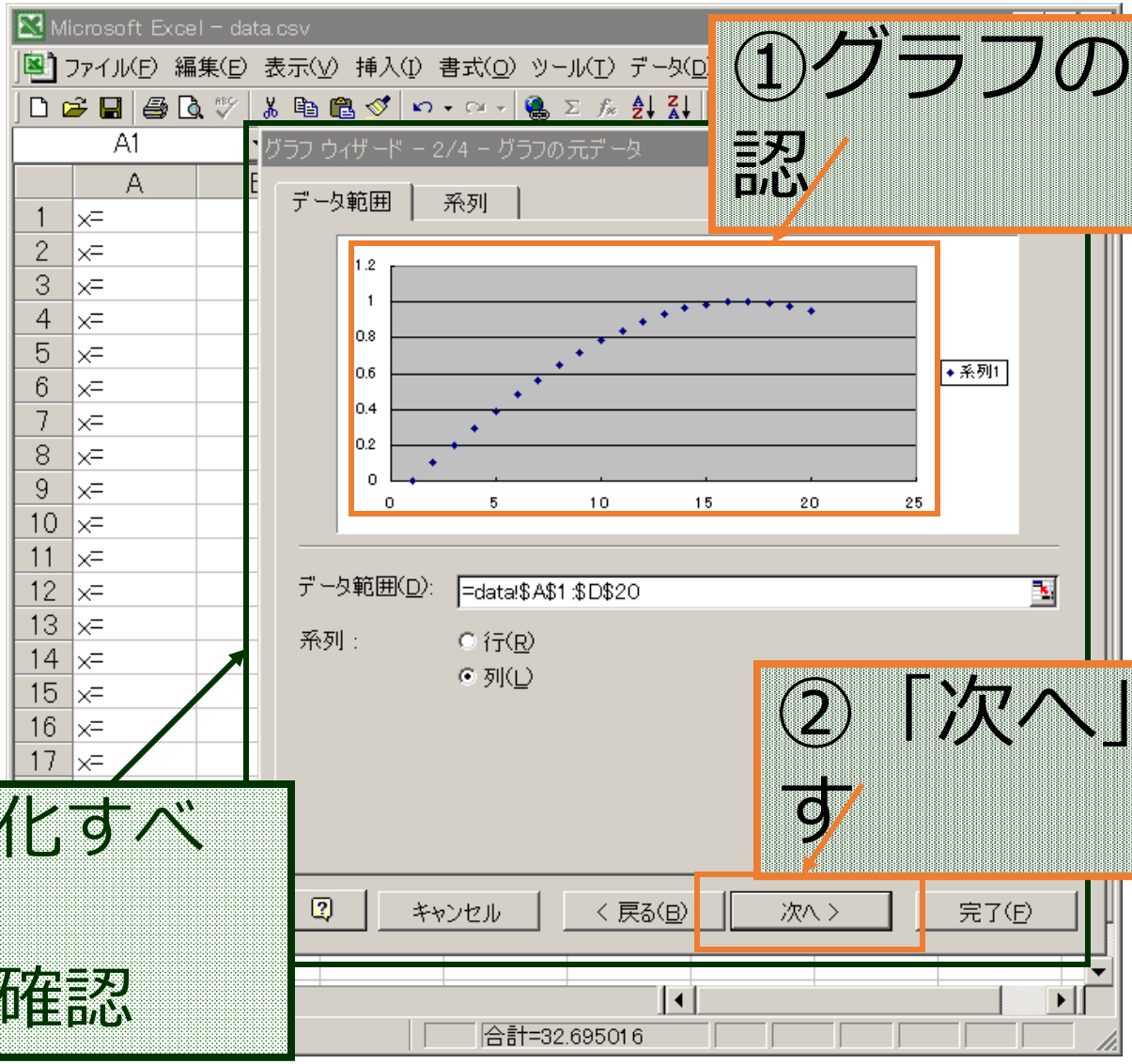
Microsoft Excel でグラフ作成 (3/8)



Microsoft Excel でグラフ作成 (4/8)



Microsoft Excel でグラフ作成 (5/8)

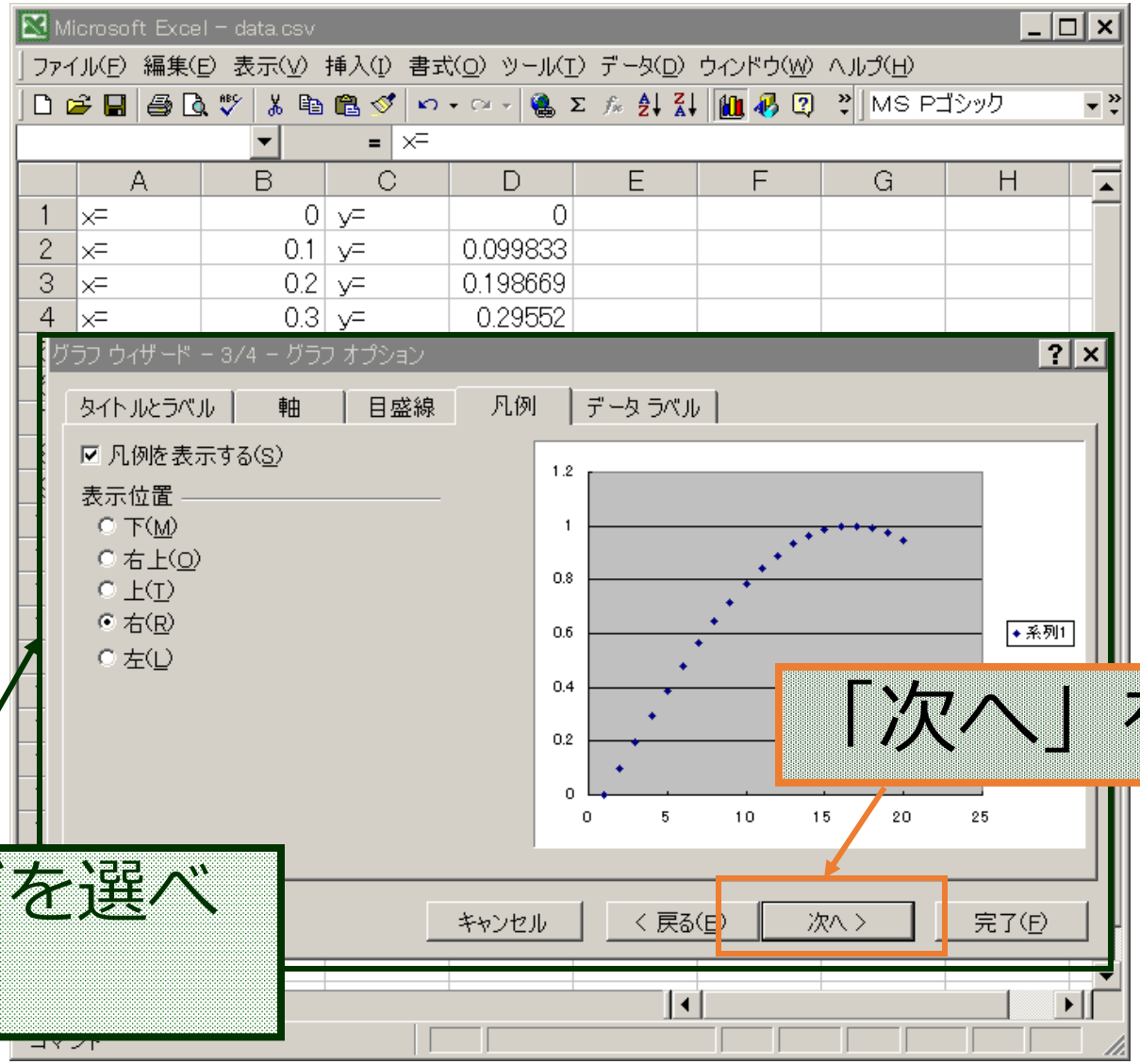


①グラフの形を確認

②「次へ」を押す

グラフ化すべき
範囲の確認

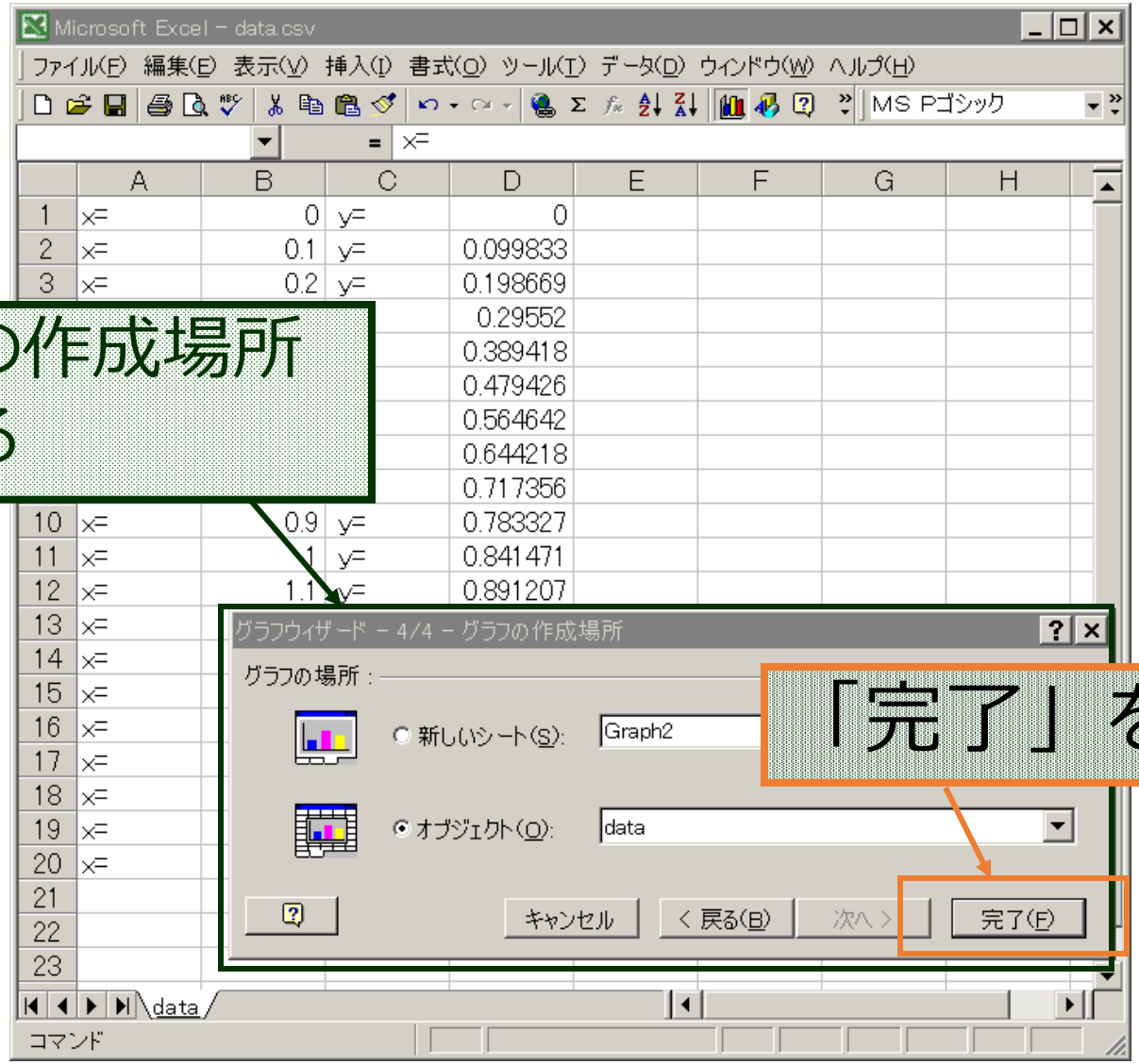
Microsoft Excel でグラフ作成 (6/8)



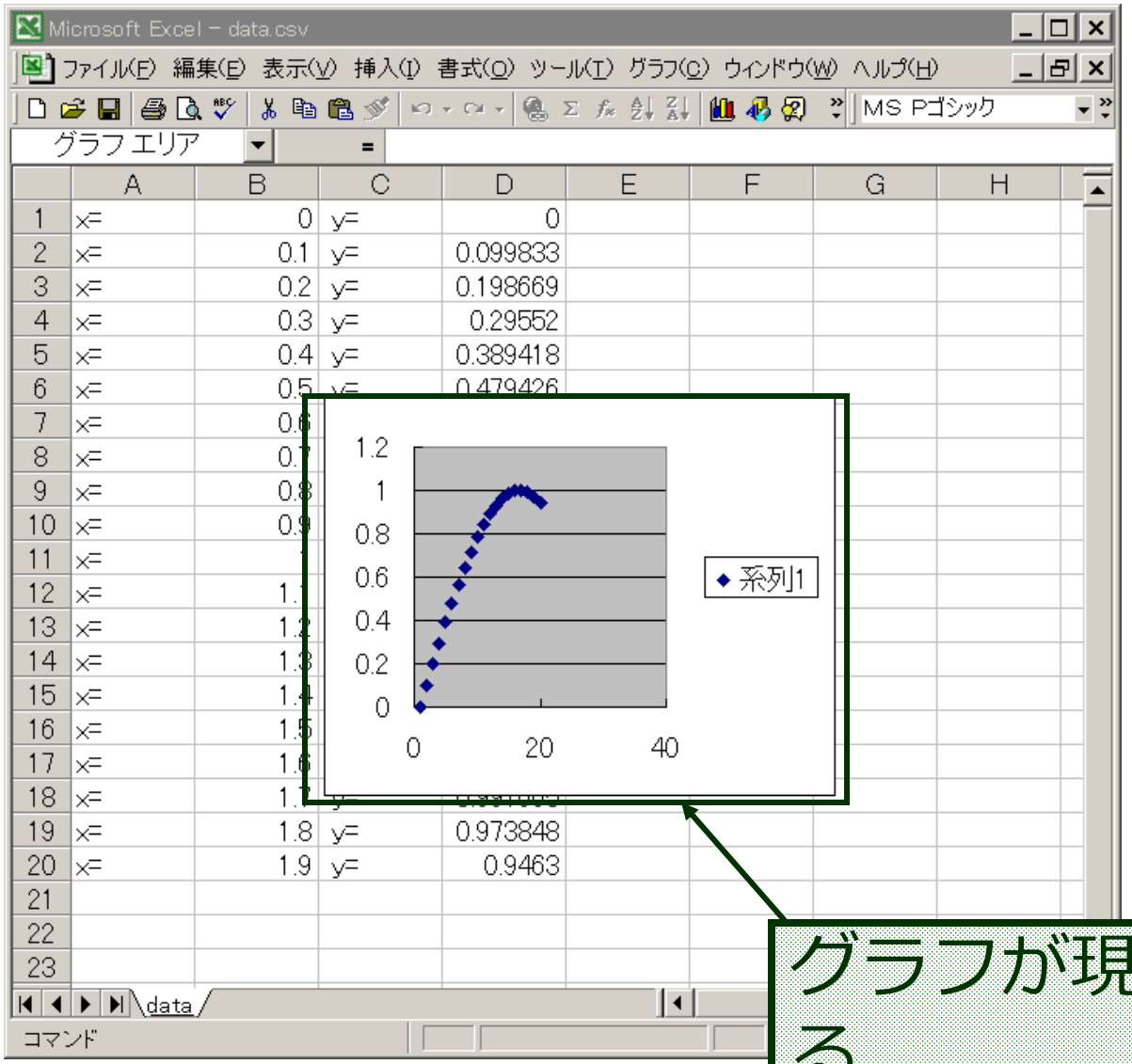
凡例などを選べる

「次へ」を押す

Microsoft Excel でグラフ作成 (7/8)



Microsoft Excel でグラフ作成 (8/8)

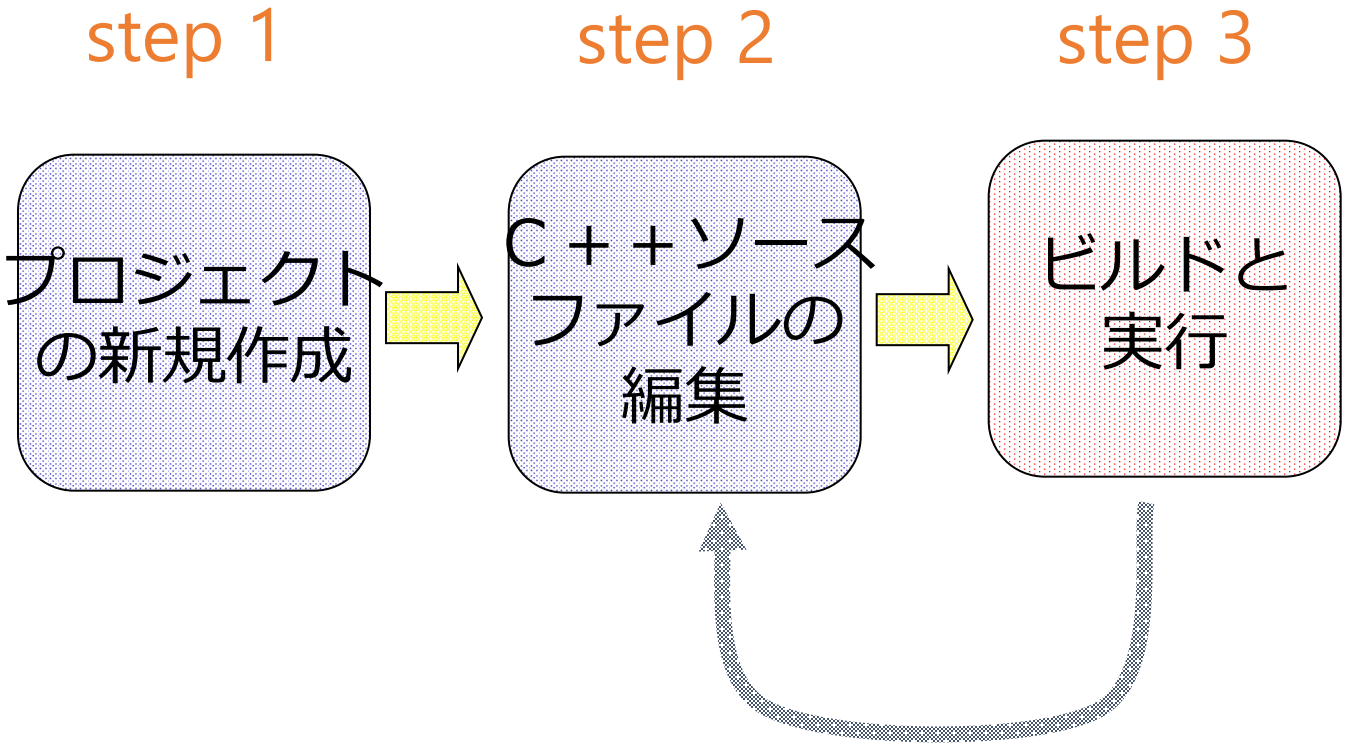


グラフが現れる

プログラムの機能

- 計算等の実行手順を記述したもの
- 多量の計算の繰り返し
  - 計算は自動で繰り返し
- キーボードからの読み込み
  - 自動で読み込み
- ファイルへの書き出し
  - ファイルを介して，他のソフトと連携
  - など

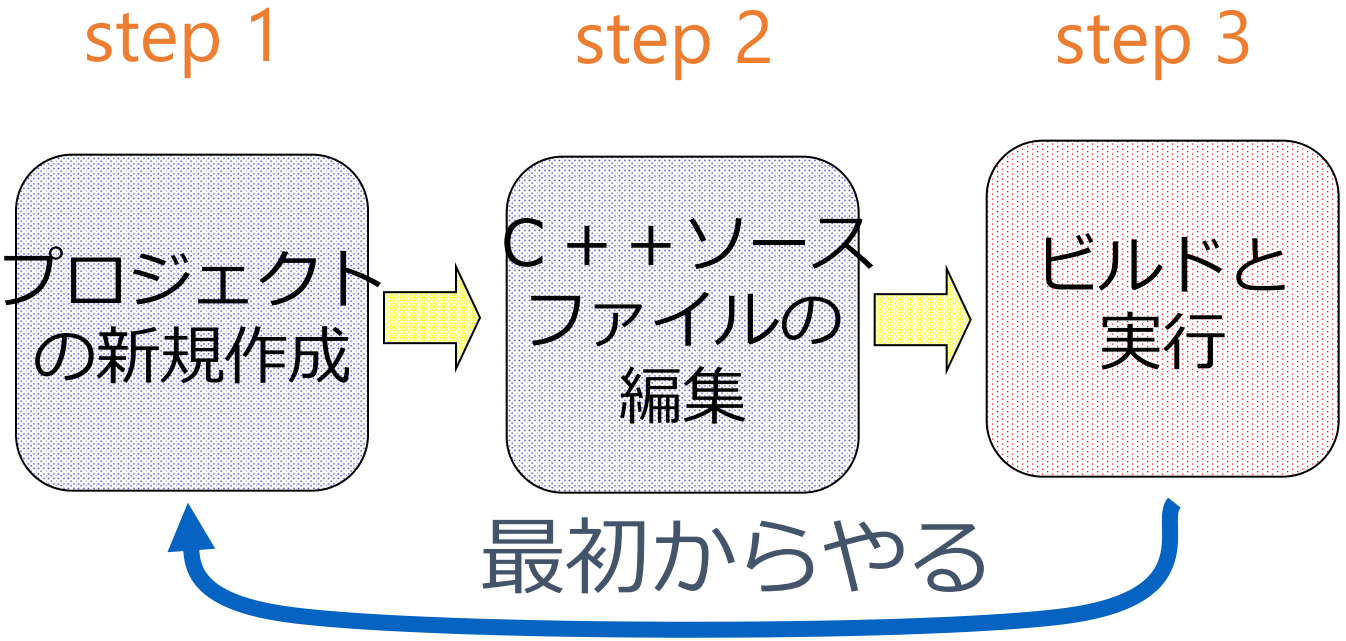
プログラム作成の流れ



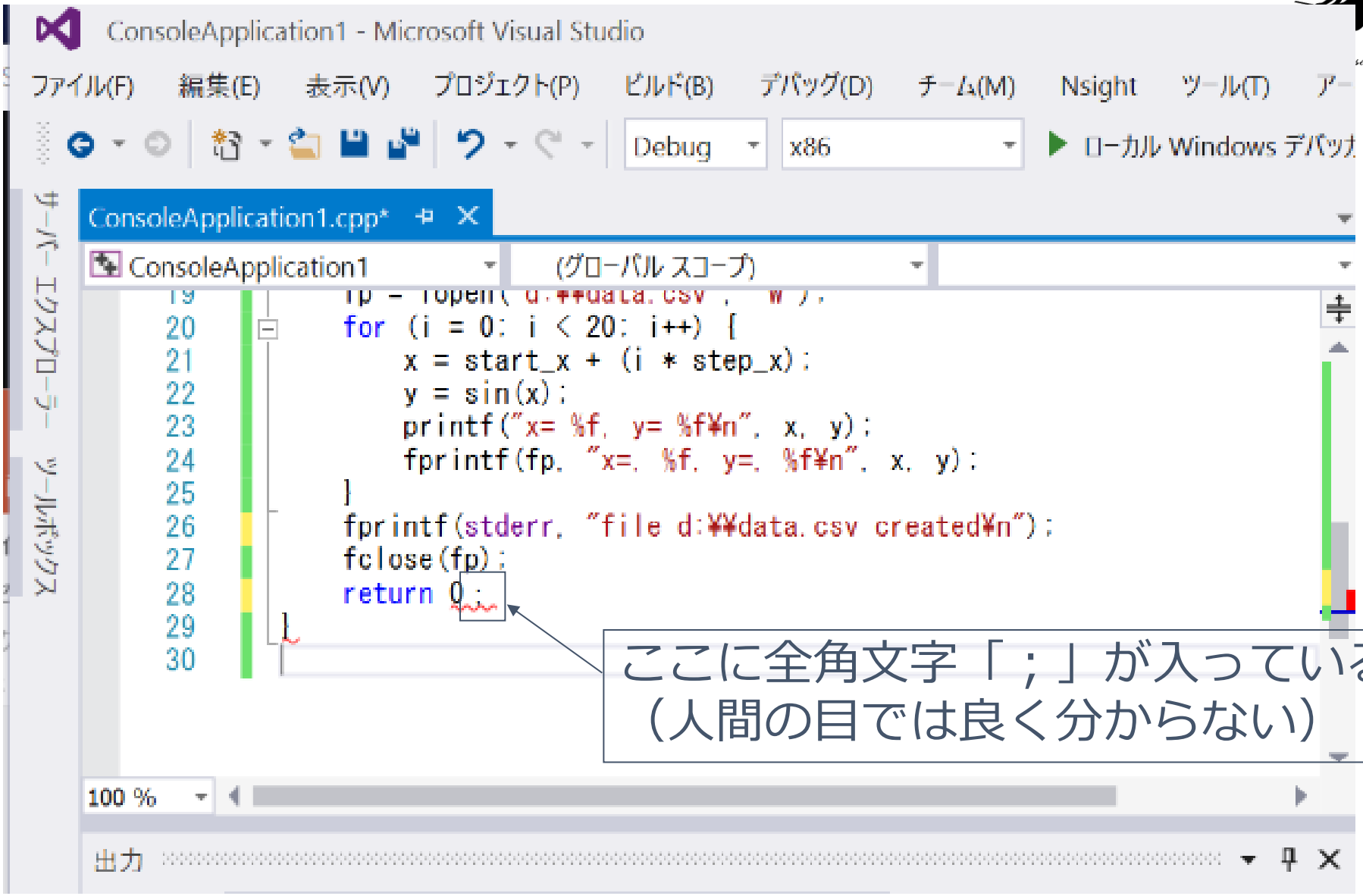
ビルド，実行の結果，間違いが見つかるたびに，繰り返す

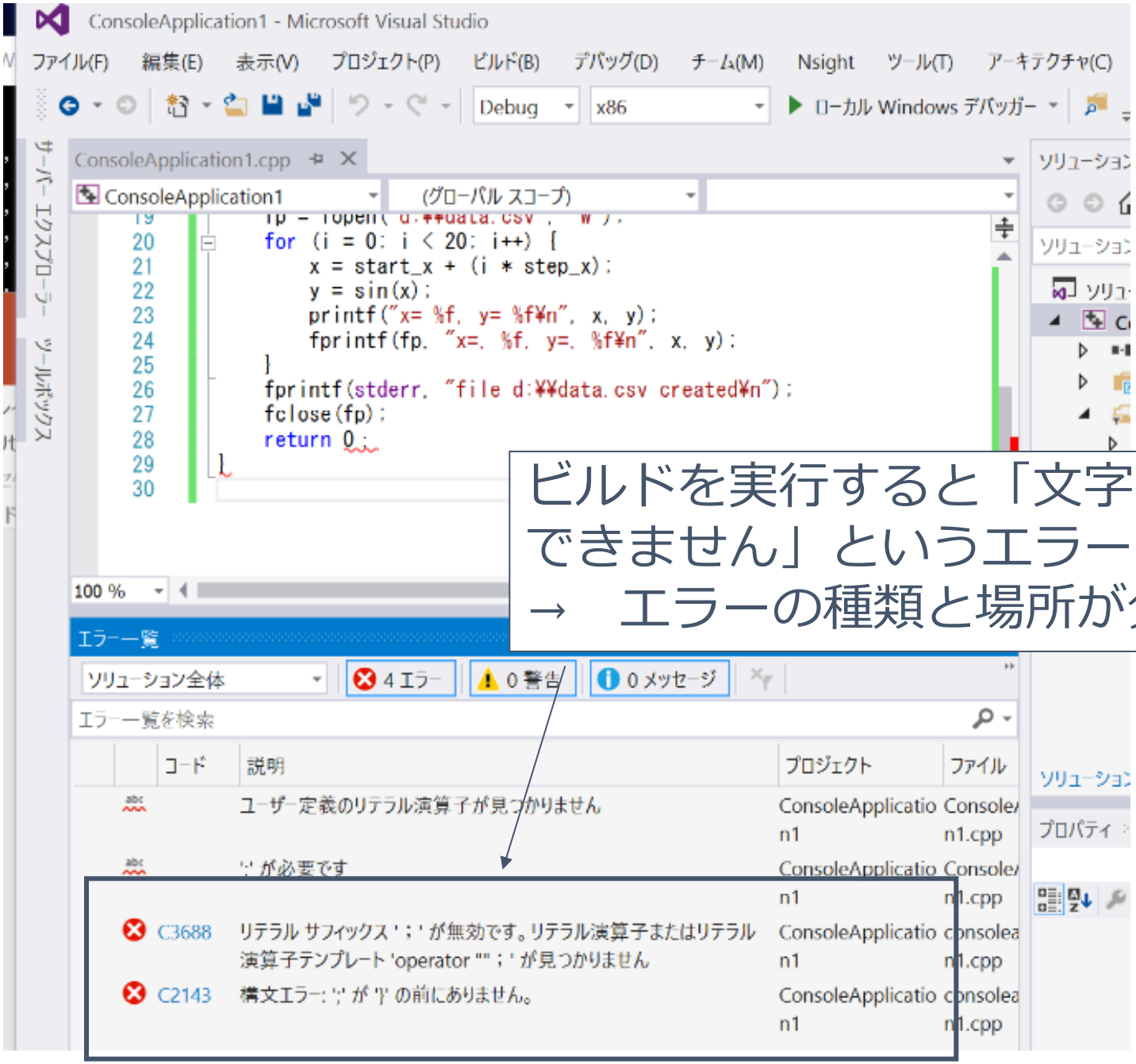
別のプログラムを作成する場合

ある課題のプログラム作成が終了し、次の課題をやる場合。
C++のソースファイルを変えるだけでは、エラーとなる。



1つのプロジェクトには、メインの関数が1つしか許されないので、新しくプロジェクトを作成する。


ConsoleApplication1 - Microsoft Visual Studio
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) Nsight ツール(T)
Debug
x86
ローカル Windows デバッガ
ConsoleApplication1.cpp*
ConsoleApplication1
(グローバル スコープ)
for (i = 0; i < 20; i++) {
x = start_x + (i * step_x);
y = sin(x);
printf("x= %f, y= %f¥n", x, y);
fprintf(fp, "x=, %f, y=, %f¥n", x, y);
}
fprintf(stderr, "file d:¥¥data.csv created¥n");
fclose(fp);
return 0;
}
ここに全角文字「；」が入っている
（人間の目では良く分からない）
100 %
出力



ビルドを実行すると「文字'0x81'は認識できません」というエラーメッセージが出
→　エラーの種類と場所が分かる

課題

課題１．Microsoft Visual Studio C++ でのプログラム作成と実行

次のプログラム（例題１のプログラム）について，例題１と例題２の手順を自分で行いなさい

```
#include "stdio.h"
#include <math.h>
int main()
{
      double x;
    double y;
    char buf[256];
    int i;
    double start_x;
    double step_x;
    FILE* fp;
    printf( "start_x =" );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf¥n", &start_x );
    printf( "step_x =" );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf¥n", &step_x );
    fp = fopen( "d:¥¥data.csv", "w" );
    for( i = 0; i < 20; i++ ) {
        x = start_x + ( i * step_x );
        y = sin( x );
        printf( "x= %f, y= %f¥n", x, y );
        fprintf( fp, "x=, %f, y=, %f¥n", x, y );
    }
    fprintf( stderr, "file d:¥¥data.csv created¥n" );
    fclose( fp );
    return 0;
}
```

データファイル名

d:¥¥data.csv
は適切に設定すること

補足説明資料

例題のプログラムの機能

- キーボードからのデータ読み込み
- 型のキャスト
- 整数の範囲で四則演算
- 結果の画面表示

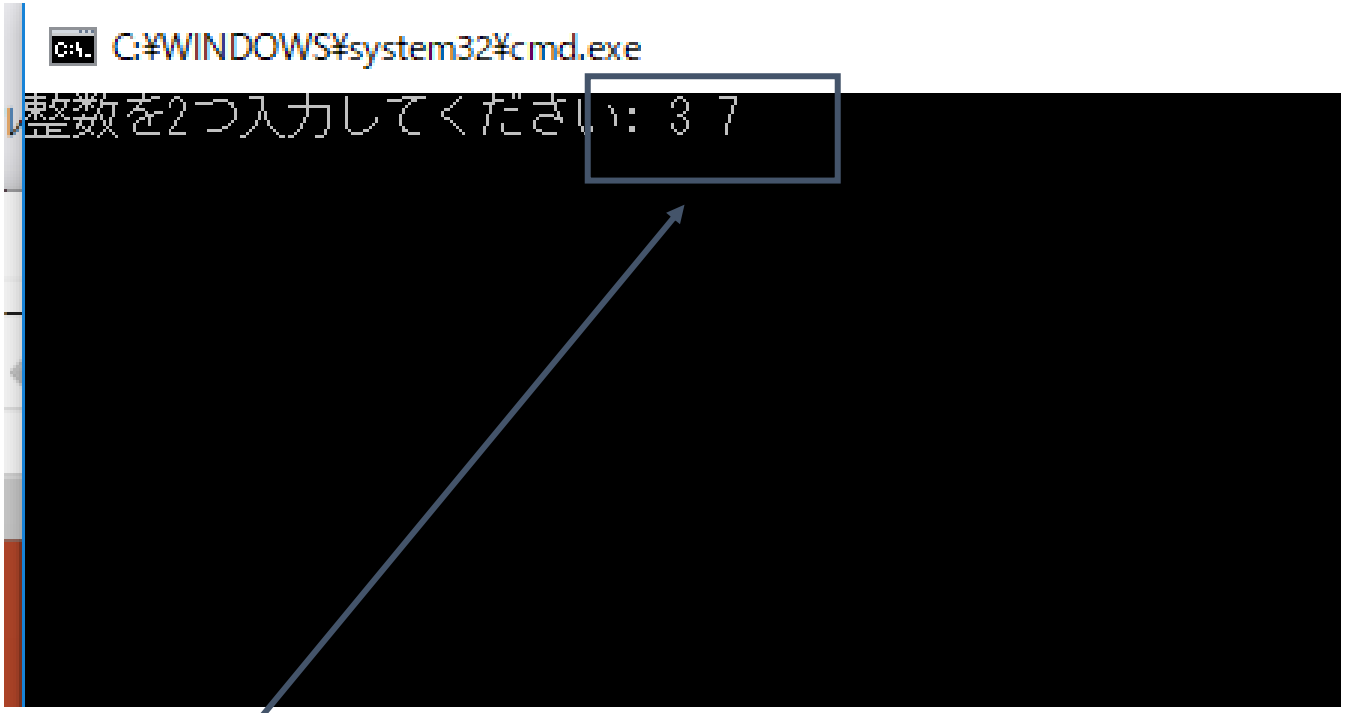
例題プログラム

```
#include "stdio.h"
void main()
{
  int i,j,k;   /* 変数宣言 */
  float f0,f1;
  printf("整数を2つ入力してください: ");
  scanf_s("%d %d", &i, &j); /*キーボードから読み込み*/
  k = i + j;   /* 算術演算 */
  f0 = (float)(i/j);          /* キャスト */
  f1 = (float)i / (float) j;
  printf("%5d + %5d = %5d¥n", i,j,k);
  printf("%5d - %5d = %5d¥n", i,j,i-j);
  printf("%5d x %5d = %5d¥n", i,j,i*j);
  printf("%5d / %5d = %5d¥n", i,j,i/j);
  printf("(float)(%d / %d) = %5.3f¥n", i,j,f0);
  printf("(float)%d / (float)%d = %5.3f¥n", i,j,f1);
}
```

ビルドと実行(1/2)

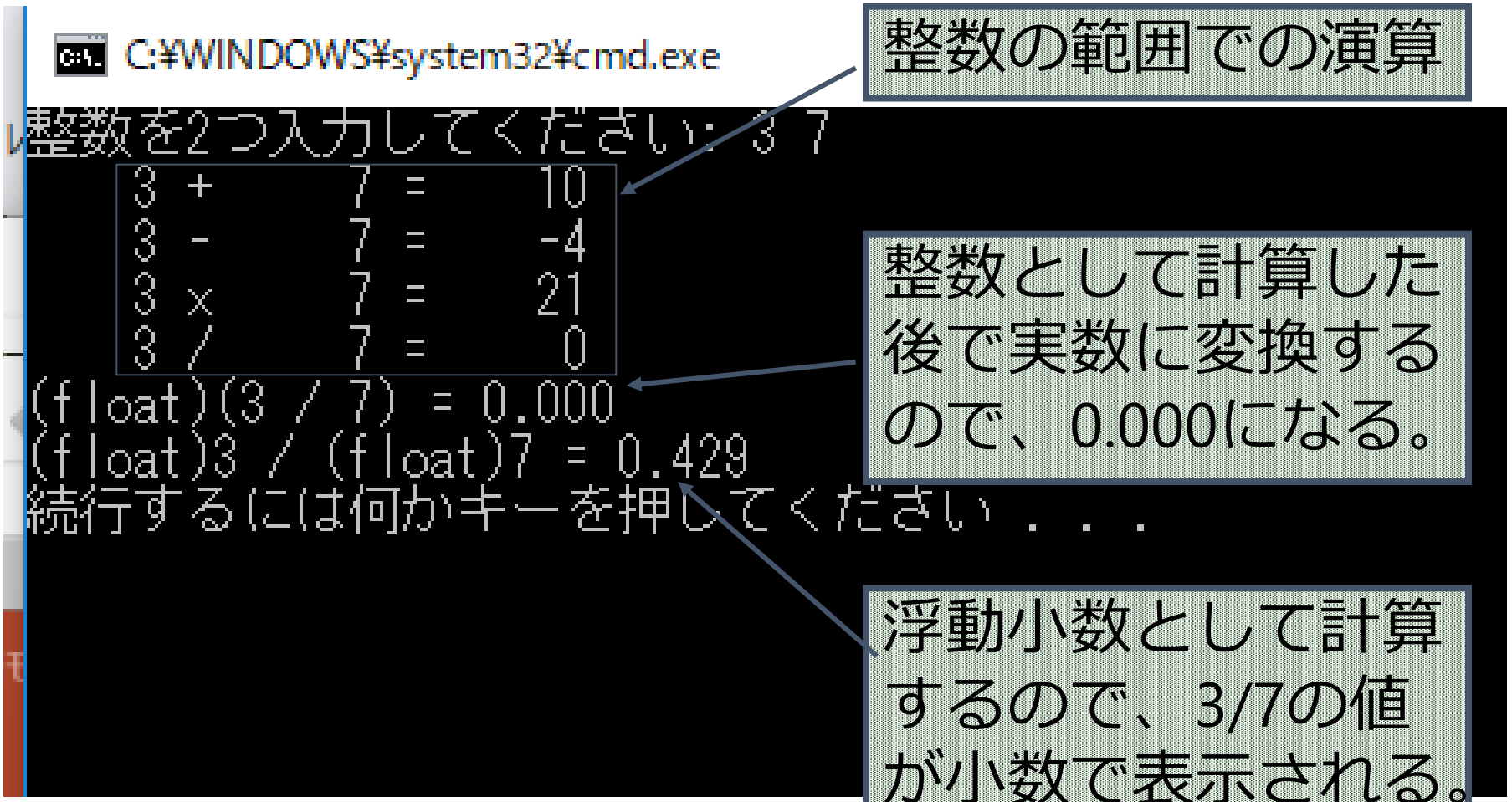
実行画面が現れる

ビルドと実行(2/2)



数値を入れる
（プログラムに数値データを与える）
ここでは，「3 7 Enter」
3 と 7 は半角．間に半角のスペース．

プログラム実行結果

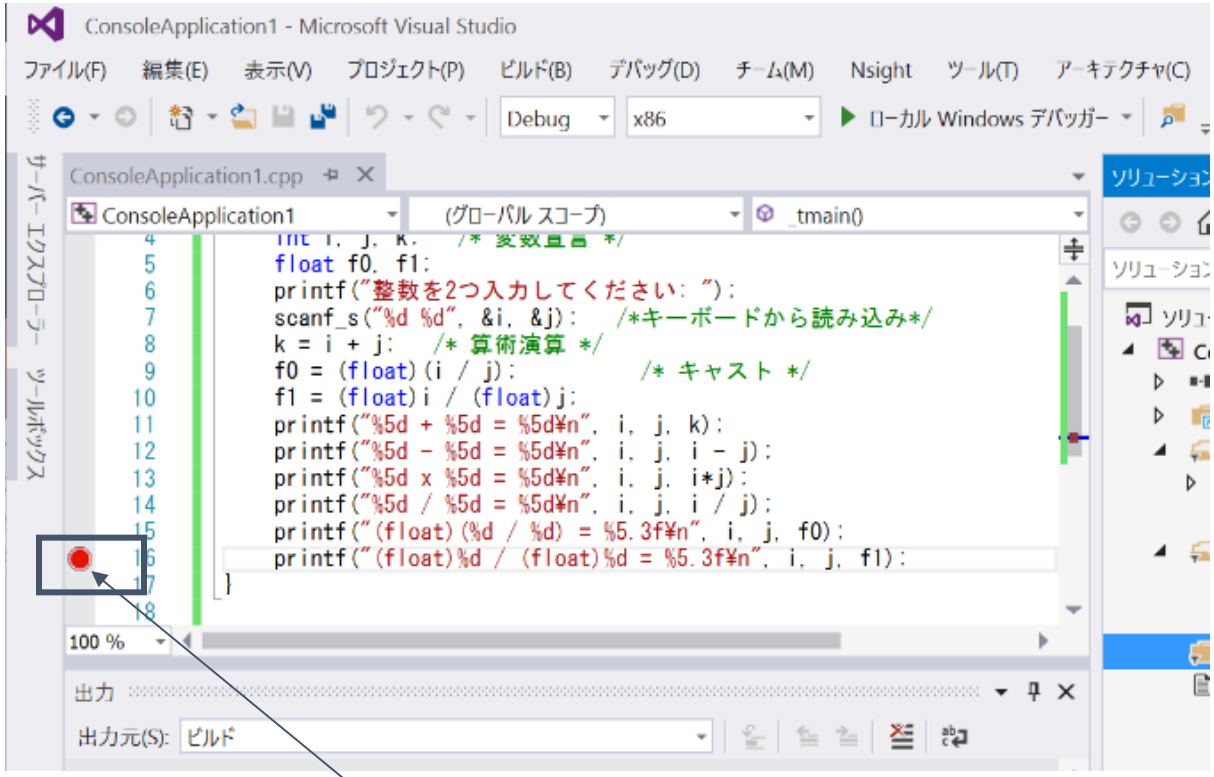


整数の範囲での演算

整数として計算した後で実数に変換するので、0.000になる。

浮動小数として計算するので、3/7の値が小数で表示される。

Microsoft Visual Studio C++ でのブレークポイント設定



ここをクリックして，ブレークポイントの設定
「●」はブレークポイント設定済みの印

ブレークポイントでプログラム実行が中断するので，
実行結果の画面が消えずに残る