



**C言語画像処理プログラミング**  
**<体験講座>**  
**プログラミングの楽しみを体験しましょう！**

講師 白石順二

# 本講座の構成

π

**事前準備：必要なプログラムのインストール**

**Step 1 Cプログラミングの環境作り**

**Step 2 パソコンへの入力と表示**

**Step 3 算数の計算をしてみましょう**

**Step 4 もしも～だったら？**

**Step 5 何度も繰り返してみましょう**

**Step 6 画像データの複製**

**Step 7 画像の階調処理**

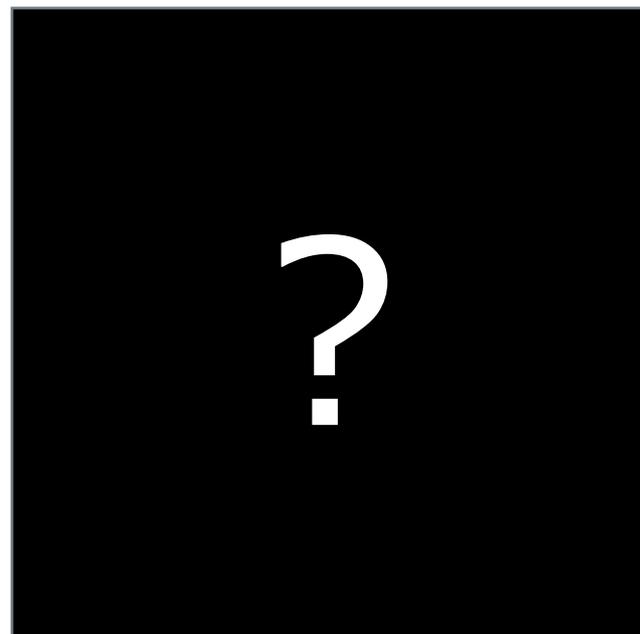
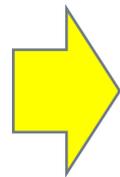
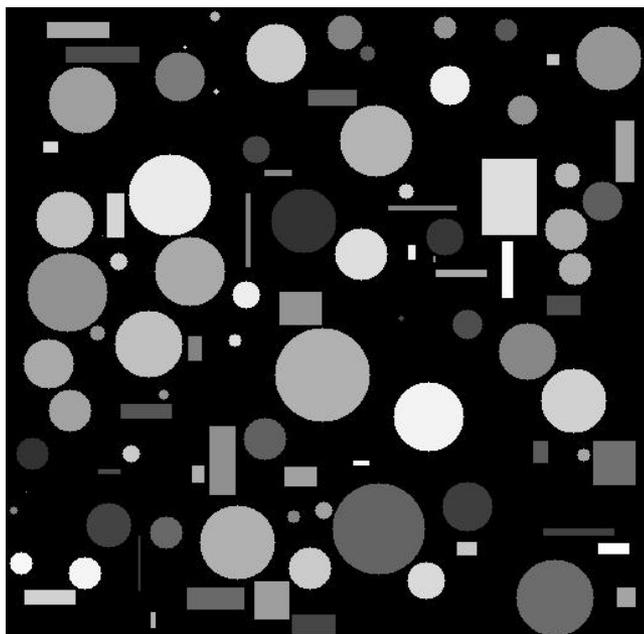
**Step 8 画像の拡大と縮小**

**Step 9 画像のヒストグラム**

**Step 10 ウォーリーを探せ！**

# 本セミナーのゴール

Cプログラミングを用いた画像処理の基礎的なことを経験し、最終的には、下に示すようなテスト画像から、ある条件に合った候補領域だけを抽出するプログラムを作成する。



# 本講座の楽しみ方

1. 提示されたソースプログラムを，その通りにソフトウェアに入力して，実行してみる.
2. ヒントにしたがって，入力したソースプログラムを変更して，再度実行し，ソースプログラムの個々の機能について学ぶ.
3. 学んだことを，ノートに記録していく.
4. 課題のプログラムを作成し，そこから得られた結果をアップロードする.
5. 課題終了ポイントを集める！10点獲得で体験講座の修了証書がもらえます.

最後まであきらめずに  
頑張りましょう！

継続は力なり！  
さあ、はじめましょう！

# 事前準備

以下に示す項目について，それぞれの資料（PDF）をダウンロードして，その指示に従ってソフトウェアをインストールし，使い方を学んでください。

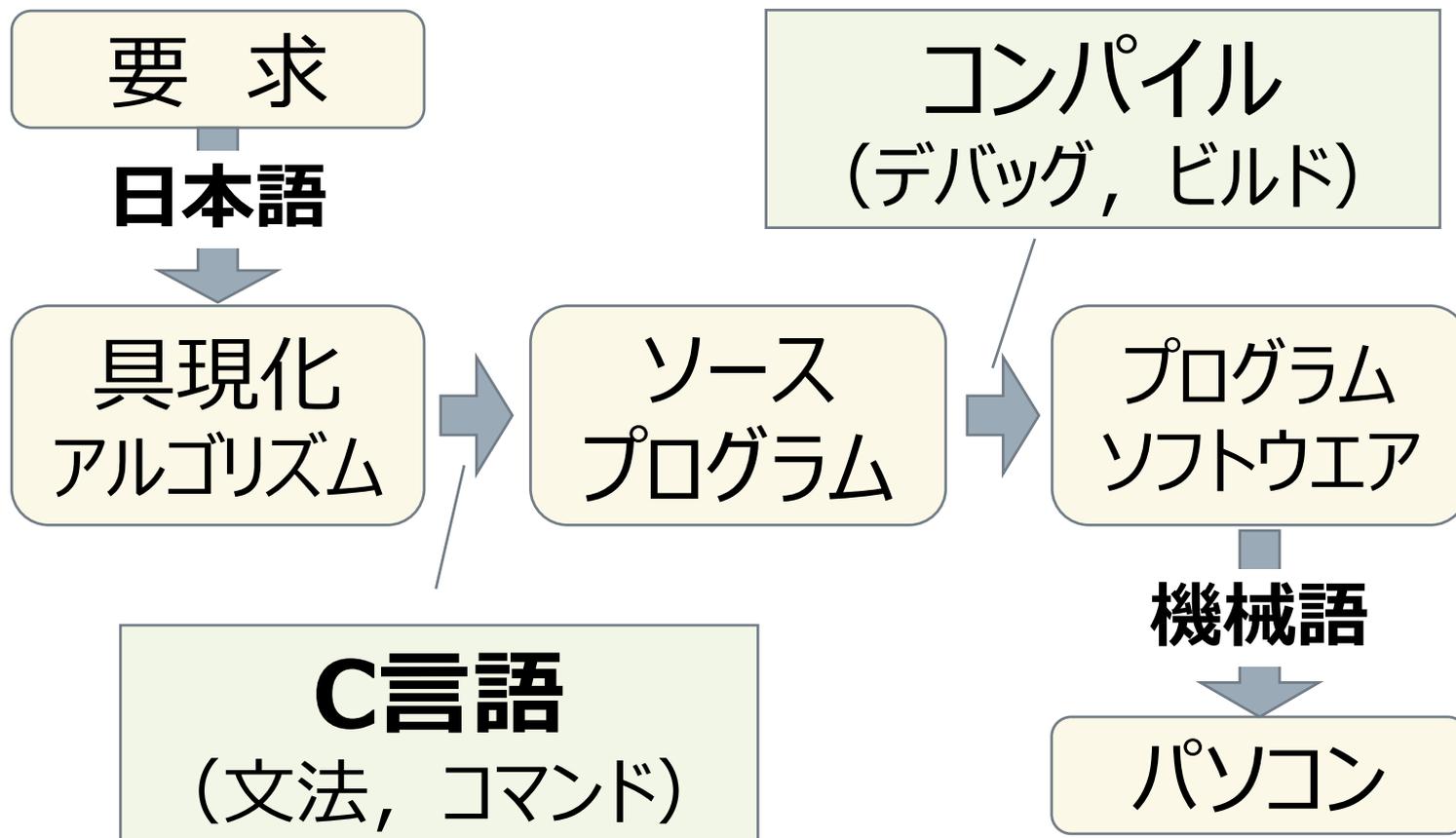
- ① Visual Studio 2019のインストール  
資料のダウンロード
- ② Visual Studio 2019の使い方  
資料のダウンロード
- ③ PowerShellの使い方  
資料のダウンロード
- ④ Image J のインストール  
資料のダウンロード



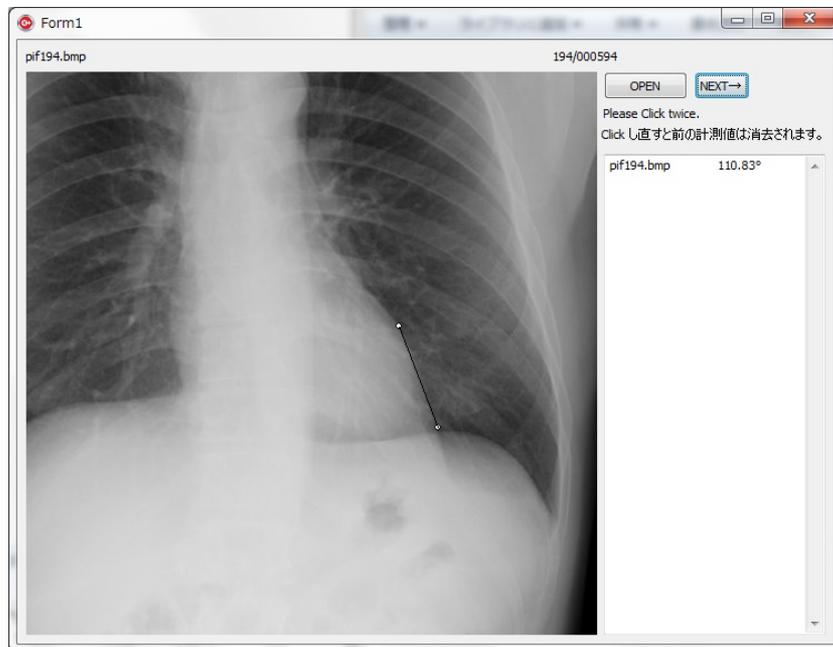
# Step 1

## Cプログラミングの環境作り

# 1-1 C言語という言語について



# 1-2 ウィンドウ型プログラムと コンソール型プログラム



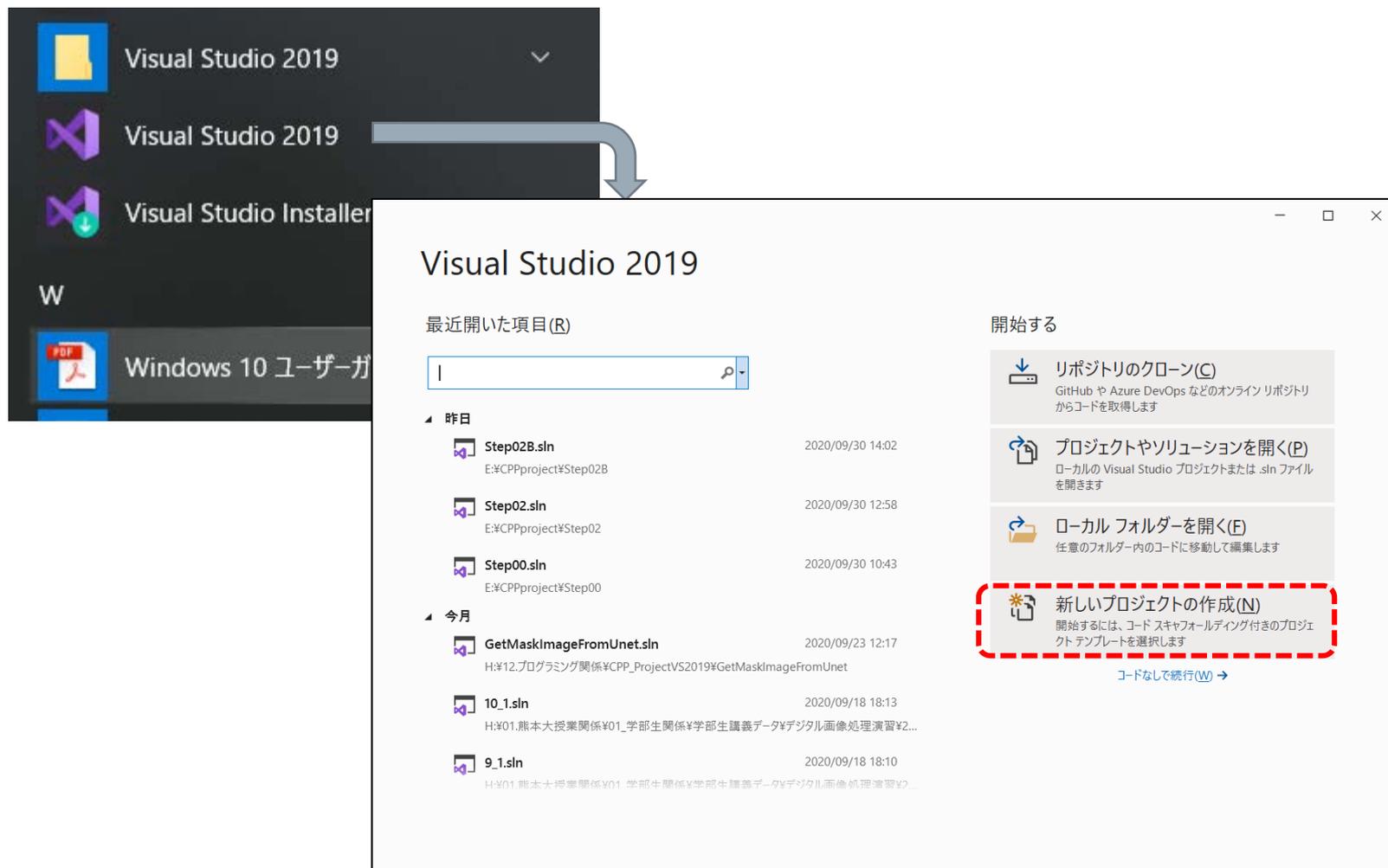
ウィンドウ型プログラム

```
コマンドプロンプト - Run10Cases.bat
No1:Area=2096904
No2:Area=3485
No3:Area=729
No4:Area=70
No5:Area=26
28 232 55 192 1 203 2096904
No. of slices in 1601516.txt:226 MatrixSize 256 X 256
D:\PIF>SegHeart.exe 1601984C-09.raw 1601984
No. of slices in 1601984C-09.raw:66
No. of planets:17
No1:Area=1695564
No2:Area=7157
No3:Area=2411
No4:Area=1111
No5:Area=985
No6:Area=890
No7:Area=300
No8:Area=258
No9:Area=225
No10:Area=162
No11:Area=132
No12:Area=98
No13:Area=88
No14:Area=55
No15:Area=31
No16:Area=1
28 227 49 175 1 209 1695564
No. of slices in 1601984.txt:241 MatrixSize 256 X 256
D:\PIF>SegHeart.exe 2007857C-09.raw 2007857
No. of slices in 2007857C-09.raw:62
```

コンソール型プログラム

# 1-3 VS Community 2019 による C言語プログラミング

## 1) 新規プロジェクトの作成 その1



The image shows a Windows Start menu on the left with 'Visual Studio 2019' and 'Visual Studio Installer' listed. An arrow points from the 'Visual Studio 2019' icon to the main application window on the right. The application window is titled 'Visual Studio 2019' and displays a '最近開いた項目 (R)' (Recently Opened Items) list and a '開始する' (Start) panel. The 'Start' panel contains three options: 'リポジトリのクローン (C)' (Clone from repository), 'プロジェクトやソリューションを開く (P)' (Open project or solution), and 'ローカル フォルダを開く (E)' (Open local folder). The '新しいプロジェクトの作成 (N)' (Create new project) option is highlighted with a red dashed border. Below it is a link 'コードなしで実行 (W) →' (Run without code).

Visual Studio 2019

最近開いた項目 (R)

検索

昨日

- Step02B.sln (E:\CPPproject\Step02B) 2020/09/30 14:02
- Step02.sln (E:\CPPproject\Step02) 2020/09/30 12:58
- Step00.sln (E:\CPPproject\Step00) 2020/09/30 10:43

今月

- GetMaskImageFromUnet.sln (H:\12.プログラミング関係\CPP\_Project\VS2019\GetMaskImageFromUnet) 2020/09/23 12:17
- 10\_1.sln (H:\01.熊本大授業関係\01\_学部生関係\学部生講義データ\デジタル画像処理演習42...) 2020/09/18 18:13
- 9\_1.sln (H:\01.熊本大授業関係\01\_学部生関係\学部生講義データ\デジタル画像処理演習42...) 2020/09/18 18:10

開始する

- リポジトリのクローン (C)  
GitHub や Azure DevOps などのオンライン リポジトリからコードを取得します
- プロジェクトやソリューションを開く (P)  
ローカルの Visual Studio プロジェクトまたは .sln ファイルを開きます
- ローカル フォルダを開く (E)  
任意のフォルダ内のコードに移動して編集します
- 新しいプロジェクトの作成 (N)**  
開始するには、コード スキャフォールディング付きのプロジェクト テンプレートを選択します

コードなしで実行 (W) →

# 1-3 VS Community 2019 による C言語プログラミング

## 1) 新規プロジェクトの作成 その1

Visual Studio 2019

Visual Studio 2019

Visual Studio Installer

W

Windows 10 ユーザーガイド

新しいプロジェクトの作成

テンプレートの検索(S) (Alt+S)

すべてクリア(C)

C++

すべてのプラットフォーム(P)

すべてのプロジェクトの種類(...)

空のプロジェクト

Windows 用に C++ で最初から始めます。開始ファイルを提供しません。

C++ Windows コンソール

コンソール アプリ

Windows ターミナルでコードを実行します。既定では "Hello World" を出力します。

C++ Windows コンソール

Windows デスクトップ ウィザード

ウィザードを使用して、独自の Windows アプリを作成します。

C++ Windows コンソール デスクトップ ライブラリ

Windows デスクトップ アプリケーション

Windows 上で実行されるグラフィカル ユーザー インターフェイス付きアプリケーションのプロジェクトです。

C++ Windows デスクトップ

共有アイテム プロジェクト

共有アイテム プロジェクトは、複数のプロジェクト間でファイルを共有するために使用されます。

C++ Android iOS Linux Windows コンソール デスクトップ

戻る(B)

次へ(N)

「すべての言語」を  
C++に変更

「空のプロジェクト」  
を選択

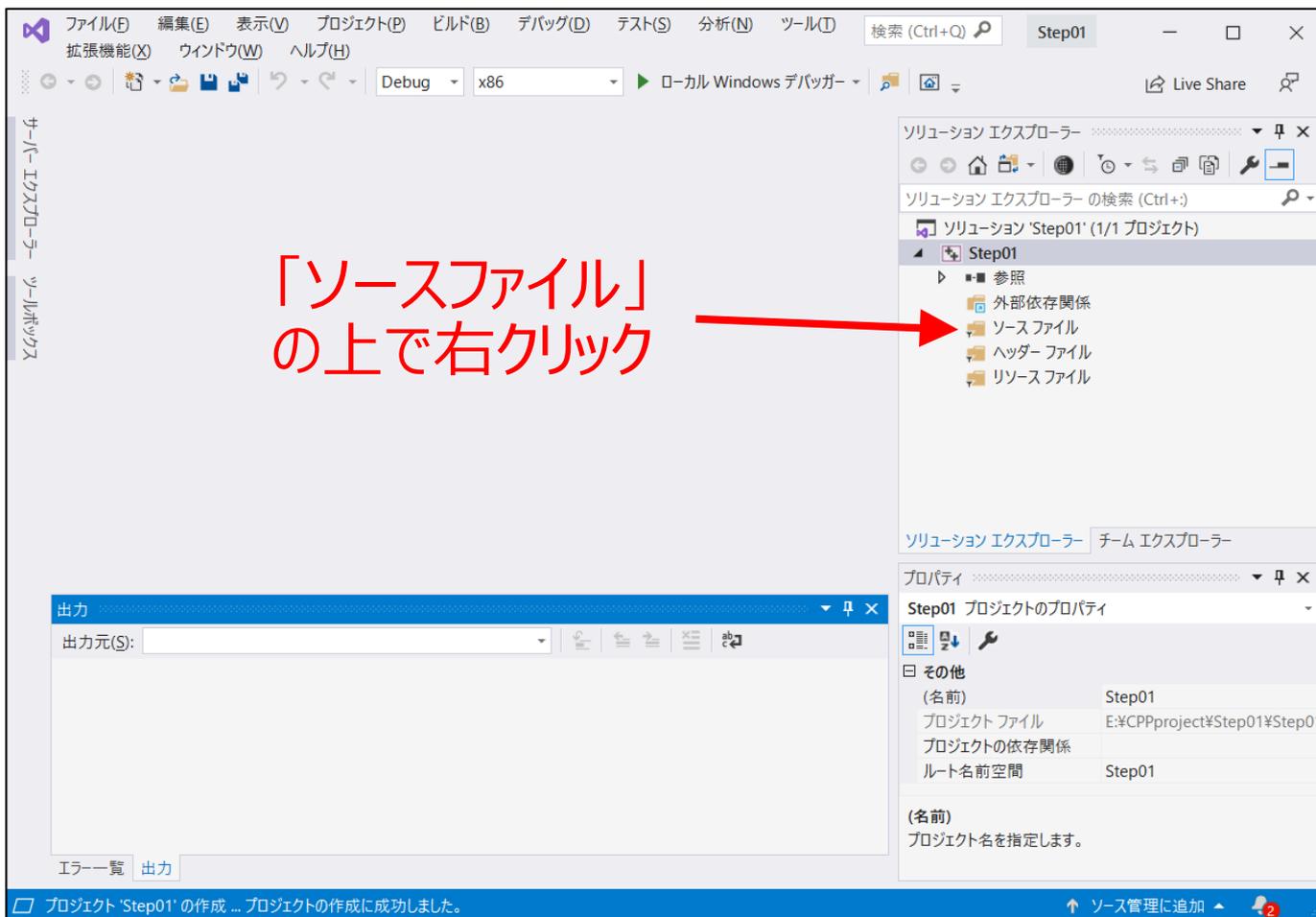
# 1-3 VS Community 2019 による C言語プログラミング

## 1) 新規プロジェクトの作成 その2



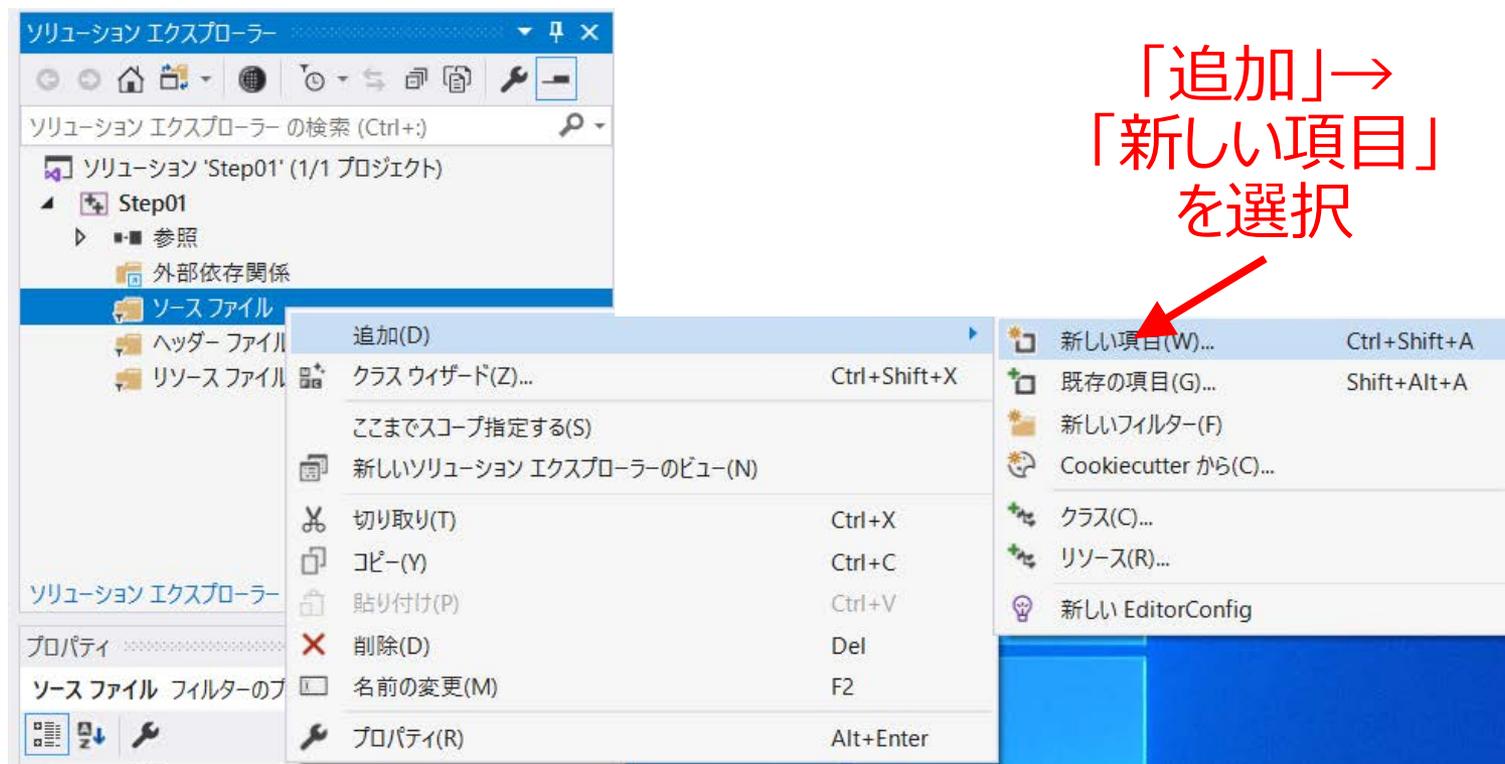
# 1-3 VS Community 2019 による C言語プログラミング

## 2) ソースプログラムの作成



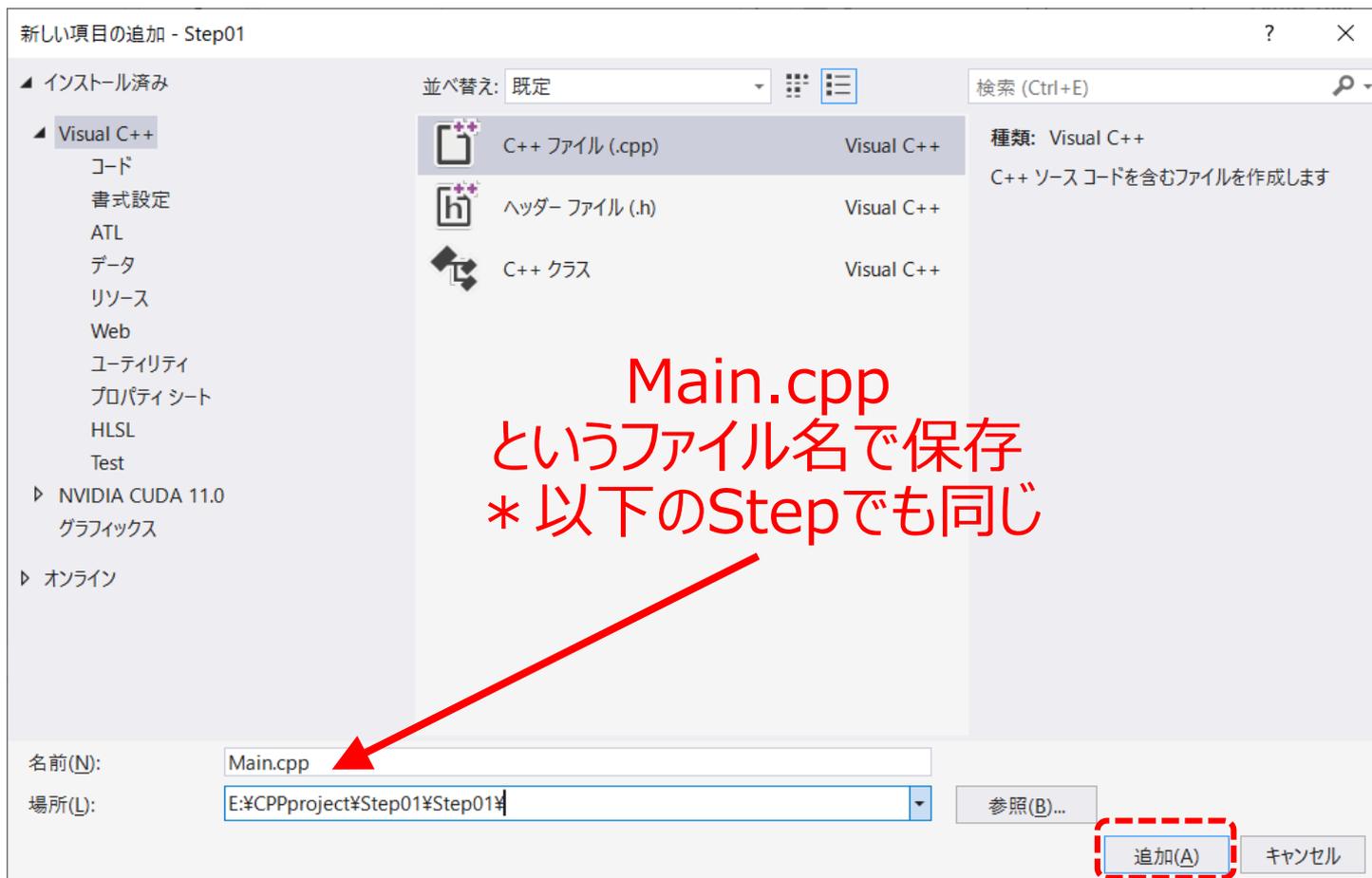
# 1-3 VS Community 2019 による C言語プログラミング

## 2) ソースプログラムの作成



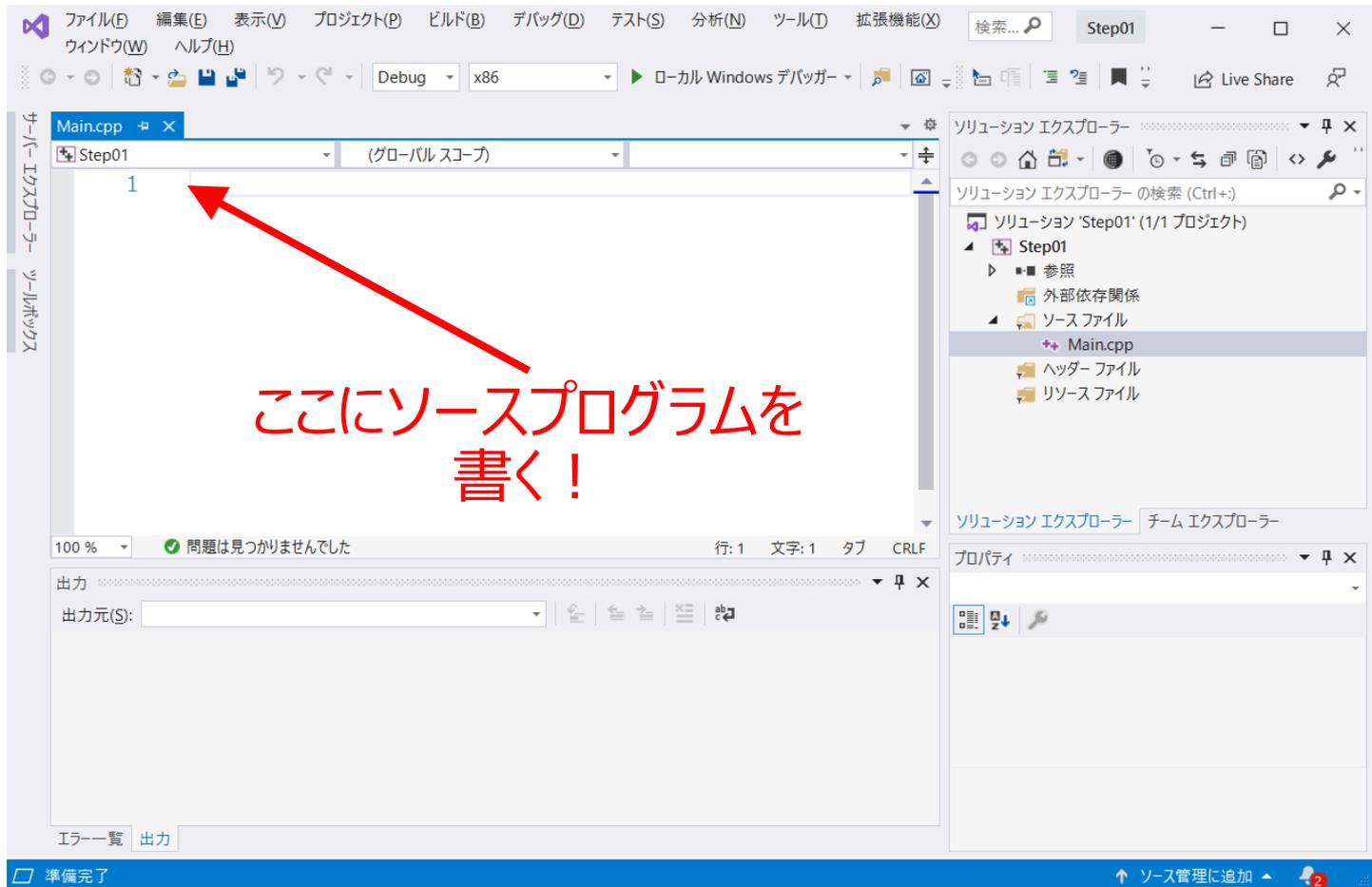
# 1-3 VS Community 2019 による C言語プログラミング

## 2) ソースプログラムの作成



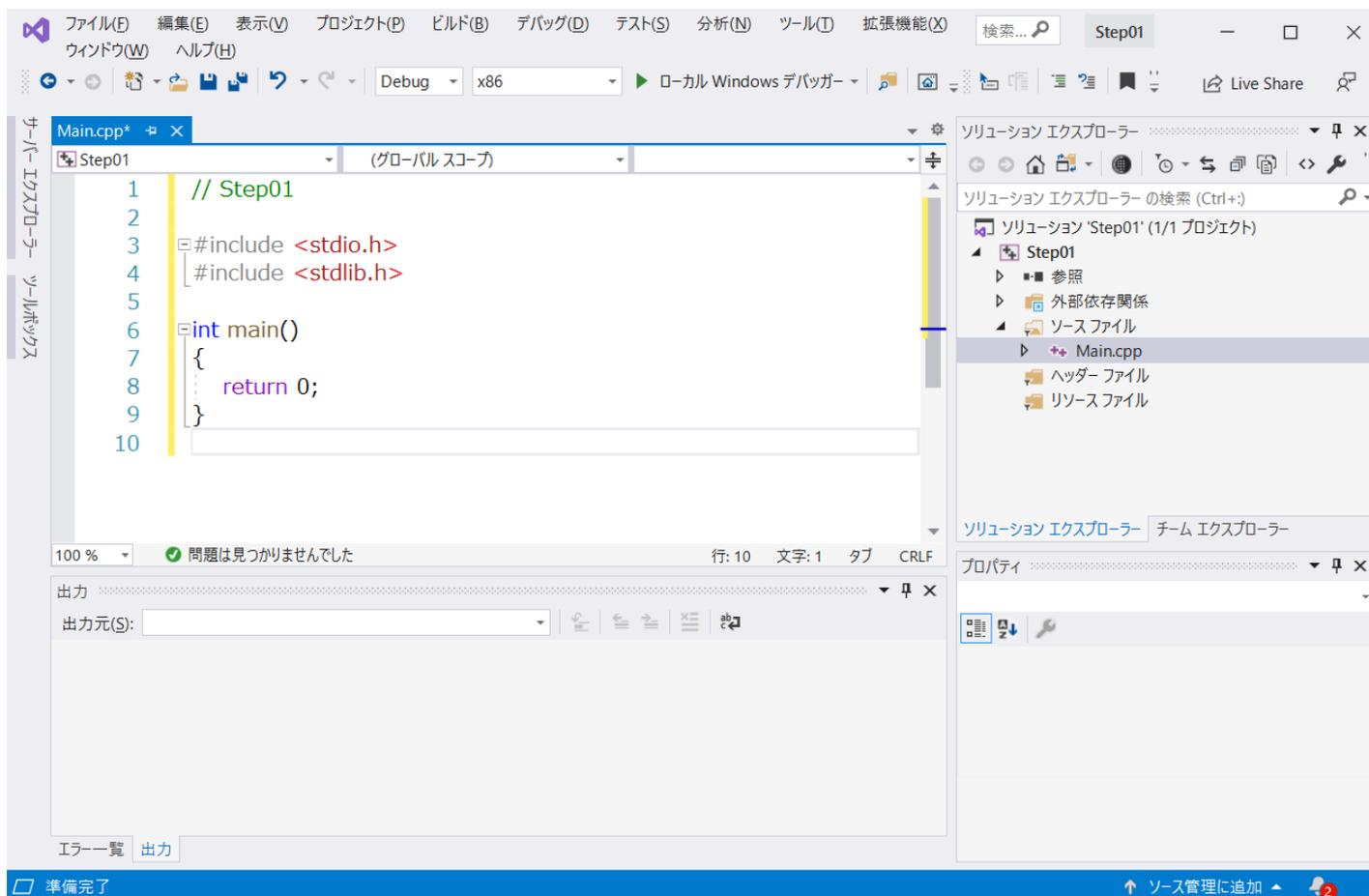
# 1-3 VS Community 2019 による C言語プログラミング

## 2) ソースプログラムの作成



# 1-3 VS Community 2019 による C言語プログラミング

## 2) ソースプログラムの作成

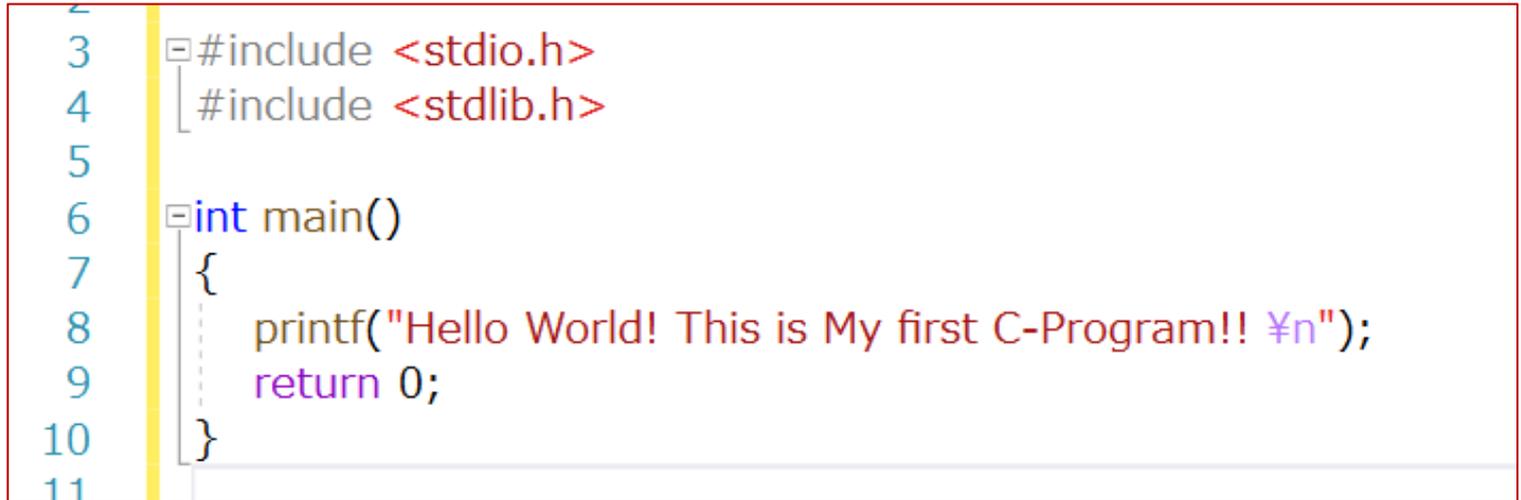


# 1-3 VS Community 2019 による C言語プログラミング

## 2) ソースプログラムの作成 その2

```
#include <stdio.h>
#include <stdlib.h>
```

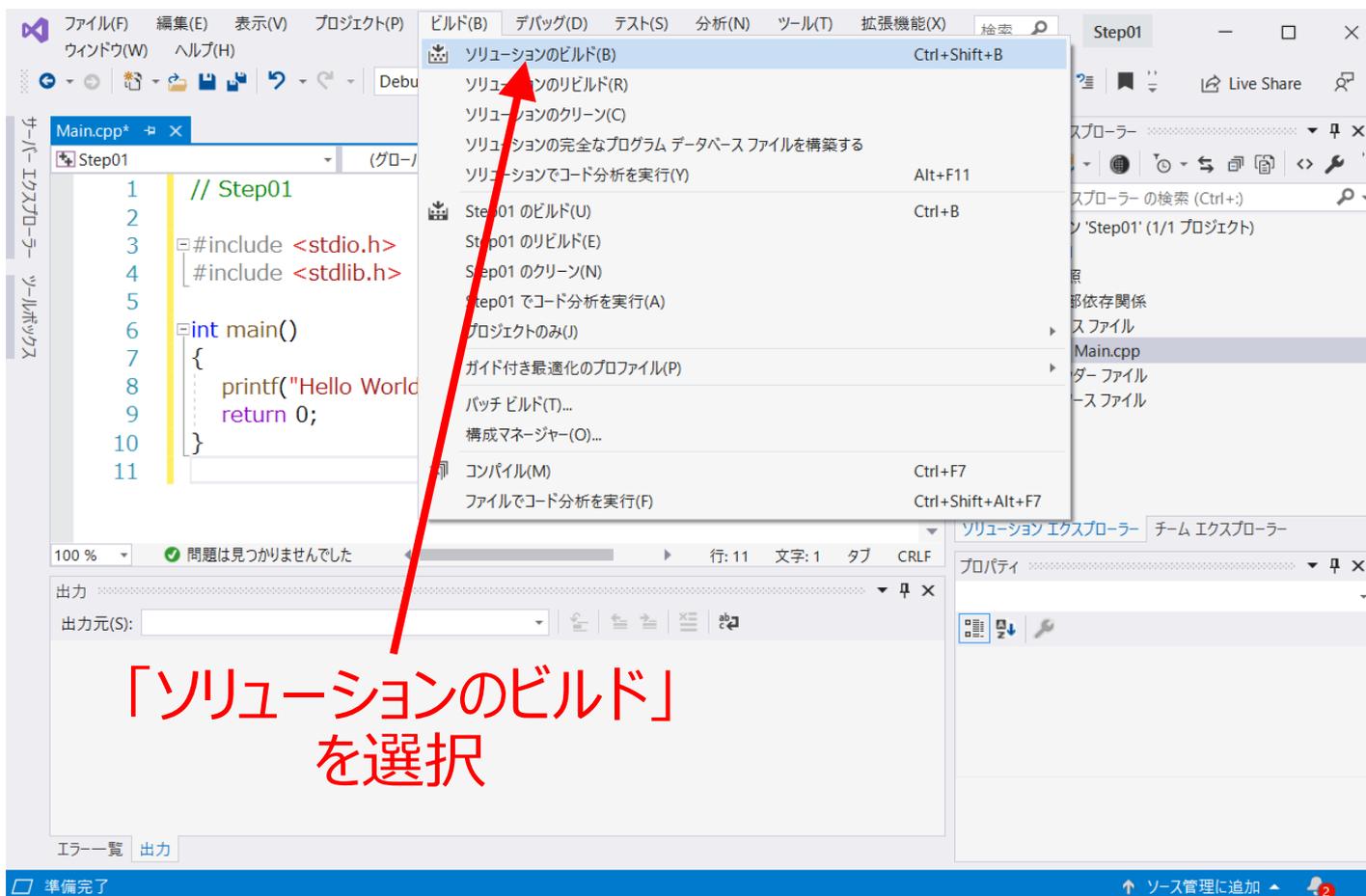
```
int main()
{
    printf ("Hello World! This is My first C-Program!! ¥n");
    return 0;
}
```



```
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 int main()
7 {
8     printf("Hello World! This is My first C-Program!! ¥n");
9     return 0;
10 }
11
```

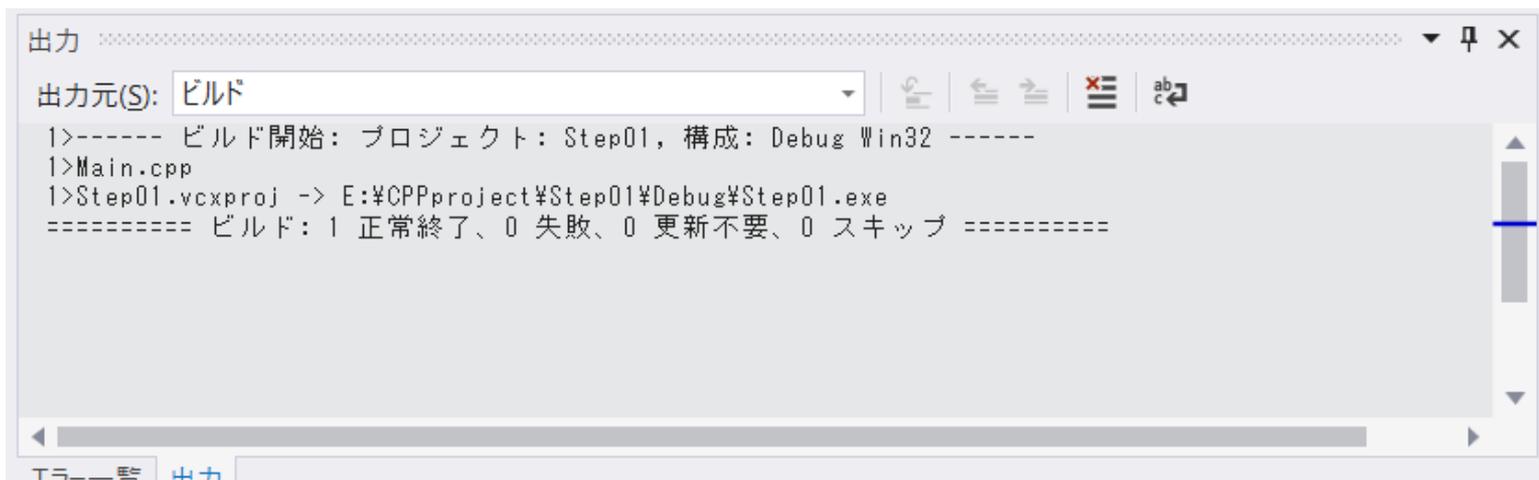
# 1-3 VS Community 2019 による C言語プログラミング

## 3) コンパイルとビルド その1



# 1-3 VS 2017 Communityによる C言語プログラミング

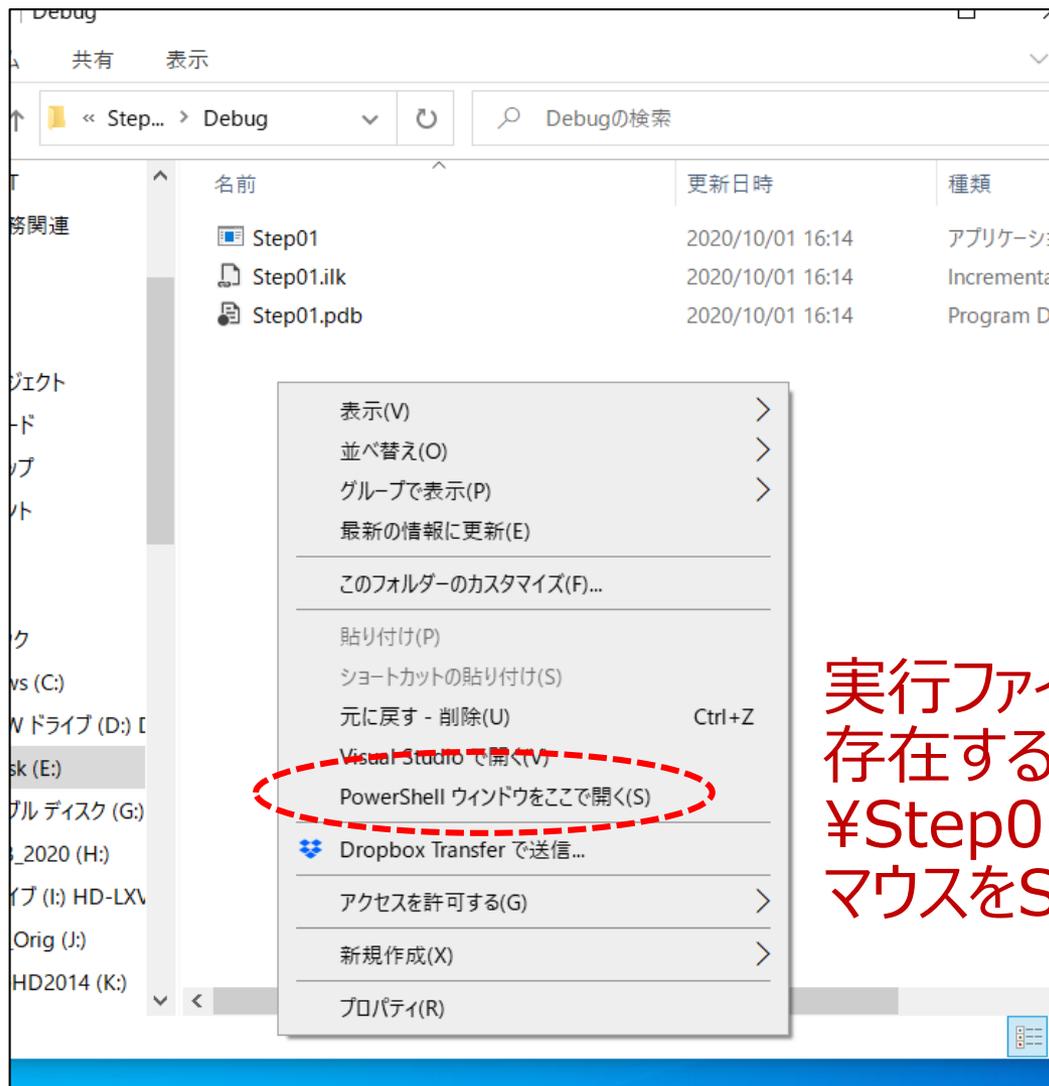
## 3) コンパイルとビルド その2



The screenshot shows the Output window in Visual Studio 2017 Community. The output source is set to 'ビルド' (Build). The output text indicates a successful build for a project named 'Step01' in 'Debug Win32' configuration. The build process includes compiling 'Main.cpp' and generating 'Step01.exe' in the path 'E:\CPPproject\Step01\Debug\Step01.exe'. The final output shows 'ビルド: 1 正常終了、0 失敗、0 更新不要、0 スキップ' (Build: 1 successful, 0 failed, 0 updates not required, 0 skipped).

```
出力
出力元(S): ビルド
1>----- ビルド開始: プロジェクト: Step01, 構成: Debug Win32 -----
1>Main.cpp
1>Step01.vcxproj -> E:\CPPproject\Step01\Debug\Step01.exe
===== ビルド: 1 正常終了、0 失敗、0 更新不要、0 スキップ =====
```

# 1-4 コマンドプロンプト (PowerShell) 上でのプログラムの実行



実行ファイルStep01.exeが  
存在するフォルダ  
¥Step01¥Debug¥内で  
マウスをShift+右クリック

# 1-4 コマンドプロンプト (PowerShell) 上でのプログラムの実行

The image shows three overlapping Windows PowerShell windows. The top window displays the command `dir` and its output, including a table of files in the directory. The middle window shows the command `./Step01.exe` being executed. The bottom window shows the same command being executed, resulting in the output "Hello World! This is My first C-Program!!".

```
Windows PowerShell
PS E:\¥CPPproject¥Step01¥Debug> dir

ディレクトリ: E:\¥CPPproject¥Step01¥Debug

Mode                LastWriteTime         Length Name
----                -
-a-----          2020/10/01   16:14           28912 Step01.exe

Windows PowerShell
PS E:\¥CPPproject¥Step01¥Debug> ./Step01.exe

Windows PowerShell
PS E:\¥CPPproject¥Step01¥Debug> ./Step01.exe
Hello World! This is My first C-Program!!
PS E:\¥CPPproject¥Step01¥Debug>
```

## Step 1 修了課題

次の問の（ ）の中に入る適当な語句（半角英数字のみ入力可）を解答Boxに記入してください。

- ① コマンドプロンプト上で、実行可能なプログラムの拡張子は>\*\*\*\*.( )である。
- ② コマンドプロンプト上で、現在のディレクトリ内に、OutDataというディレクトリ（フォルダ）を新規作成する際の入力は>( ) ./OutData である。
- ③ 新規作成したOutDataのディレクトリに移動する際のコマンドは>( ) ./OutData である。



## Step 2

### パソコンへの入力と出力

## 2-1 文字列の入力

Step2のプロジェクトを新規に作成し、  
以下のソースプログラムを作成してビルドしてください。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

} 今後は、この3行を常に追加！

```
int main()
{
    char buf[200];

    printf("あなたの名前を入力して下さい¥t");
    gets_s(buf, sizeof(buf));

    printf("¥nはじめまして、 %s さん。¥n", buf);
    return 0;
}
```

## 2-2 文字列の入力と表示

### プログラム内に文字列を取り込むコマンド

```
gets_s(buf, sizeof(buf));
```

キーボードから入力された文字列を、あらかじめ指定した文字数を最大として、bufに取り込む。

### PCの画面上に文字列を表示するコマンド

```
printf(" ..... ¥n");
```

"と"で囲まれた部分を画面に表示する。

¥n : 改行する。

```
printf(" ...%s..... ¥n", buf);
```

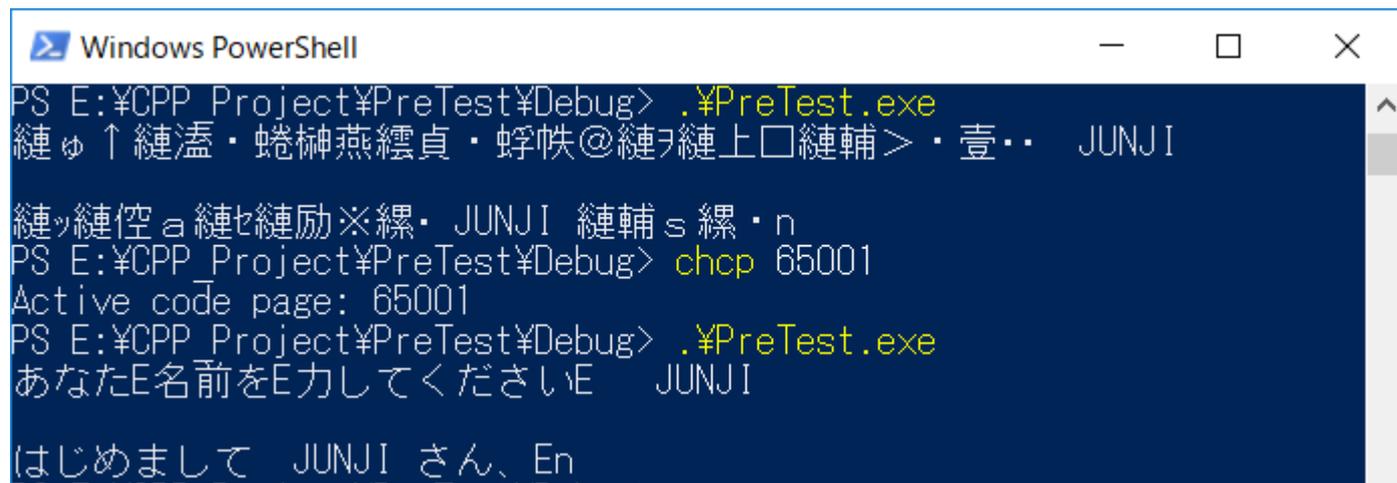
%s : bufに代入された文字列を画面に出力する。

# 文字化けしてる？

PowerShellでプログラムを実行した場合，下に示すようにプログラムで記述した日本文が文字化けすることがあります．これは，PowerShellで使用している文字コードの違いが原因ですので，

```
¥Debug> chcp 65001
```

と入力して，文字の表示コードをUTF-8に変更すると，少しの文字化けは残りますが，日本語で表示されます．文字化けが嫌な人は英語で入力！



```
Windows PowerShell
PS E:\¥CPP Project¥PreTest¥Debug> .¥PreTest.exe
縹ϕ ↑ 縹湓・蜷榭燕縹貞・蜉舛@縹ヲ縹上口縹輔> ·壹·· JUNJI

縹ッ縹空 a 縹セ縹励※縹・ JUNJI 縹輔 s 縹・n
PS E:\¥CPP Project¥PreTest¥Debug> chcp 65001
Active code page: 65001
PS E:\¥CPP Project¥PreTest¥Debug> .¥PreTest.exe
あなたE名前をE力してくださいE JUNJI

はじめまして JUNJI さん、En
```

## 2-3 文字列から数値への変換

以下のソースプログラムを作成してビルドしてください。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    char buf[200];

    printf("Would you please input your age? %t");
    gets_s(buf, sizeof(buf));
    int age = atoi(buf);

    printf("%nOh, you are %d yeas old! %n", age);
    return 0;
}
```

## 2-4 文字列から整数への変換と表示

### 取り込んだ文字列を整数に変換するコマンド

```
int x = atoi(buf);
```

bufに取り込んだ文字列を，整数(Integer: int)の変数(x)に代入する.

atoi → ASCII TO INTEGERの略

### PCの画面上に整数を表示するコマンド

```
printf(" .....%d..... ¥n", x);
```

**%d** : 変数 x に代入された整数を画面に出力する.

## 2-5 文字列から小数への変換と表示

### 取り込んだ文字列を小数に変換するコマンド

```
float a = atof(buf);
```

bufに取り込んだ文字列を，小数(Float: float)の変数(a)に代入する。

atof → ASCII TO FLOATの略

### PCの画面上に小数を表示するコマンド

```
printf(" .....%f..... ¥n", a);
```

**%f** : 変数aに代入された小数を画面に出力する。

# Cプログラミング上達のヒント！

“=”はイコールではなく，代入！

$$X = 23 ;$$

ソースプログラミングで記述された上の式の意味は，「Xは23に等しい」ではなく，「Xに23を代入する」ということ。

$$X = X + 1 ;$$

この式は，「Xに1を加える」という意味。Cプログラミングでは，常に右辺の値が左辺に代入されます。

## 2-6 printfによる変数の表示

int a=33の場合 ;	表示例
printf("a=%d",a);	a = 33
printf("a=%3d",a);	a = 33
printf("a=%4d",a);	a = 33
printf("a=%04d",a);	a = 0033

float c=33.333の場合 ;	表示例
printf("c=%f",c);	a = 33.333000
printf("c=%3.0f",c);	a = 33
printf("c=%3.2f",c);	a = 33.33
printf("c=%3.4f",c);	a = 33.3330

## Step 2 修了課題

- ① 整数の変数  $x$  に 1 が代入されている状態で、PCの画面上に以下のように表示させたい場合に、以下のソースプログラムの \* \* \* \* に入る適当な語句は何でしょうか。

表示例 :  $x = 0001$

```
printf("x = * * * * %n", x);
```

- ② 小数の変数  $a$  に 0.123456 が代入されている状態で、PC画面上に以下のように表示させたい場合、以下のソースプログラムの \* \* \* \* に入る適当な語句は何でしょうか。

表示例 :  $a = 0.12$

```
printf("a = * * * * %n", a);
```

\* どちらの設問も、自分でプログラムを作成して、正解を確認してから回答するようにしましょう。



## Step 3

算数の計算をして見ましょう

## 3-1 変数の宣言と四則演算

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>    // 数値計算のライブラリ

int main()
{
    int a, b; // aとbをint型の整数として宣言
    float c, d; // cとdをfloat型の浮動小数点として宣言

    a = 12;          b = 4;
    c = 3.12;        d = -12.3;

    printf("a+b=%d¥n",a+b);
    printf("c-d=%f¥n", c-d);
    return 0;
}
```

# Cプログラミング上達のヒント！

コメント文を多用しましょう！

```
// aとbをint型の整数として宣言
```

ソースプログラミング内で"/"の後ろに記述された内容は、プログラムの実行には関係がない文章でコメント文と呼ばれます。後でソースプログラムを見直す場合や、第3者が見てもわかるように、ソースプログラムではコメント文を多用しましょう。

## 3-2 対話型の数値計算プログラム

以下のソースプログラムを作成してビルドしてください。

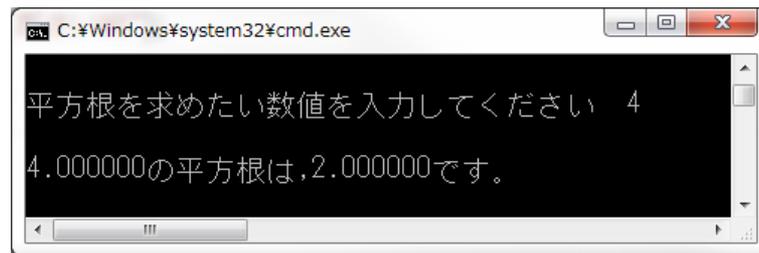
```
int main()
{
    char buf[200];
    float c, d;

    printf("¥n平方根を求めたい数値を入力してください¥t");
    gets_s(buf, sizeof(buf));

    c = atof(buf);
    d = sqrt(c);    // cの平方根を計算する

    printf("¥n%fの平方根は,%fです。¥n", c, d);
    return 0;
}
```

出力例



```
C:\Windows\system32\cmd.exe
平方根を求めたい数値を入力してください 4
4.000000の平方根は,2.000000です。
```

## 3-3 数値計算の記号とコマンド

記号・コマンド	意味	使用例	Xに代入される値
+	足し算	$X = 3 + 4;$	7
-	引き算	$X = 6.2 - 5.0;$	1.2
*	掛け算	$X = 2.3 * 2.0;$	4.6
/	割り算	$X = 6.0/2.0;$	3.0
%	剰余	$X = 100\%3;$	1 *1
sqrt	平方根	$X = \text{sqrt}(9.0);$	3.0 *2
pow	累乗	$X = \text{pow}(3.0,2.0);$	9.0 *3

\*1 : 100を3で割った余り. 計算可能なのは整数のみ

\*2 : 9.0の平方根 ( ) の中の値は必ず浮動小数点

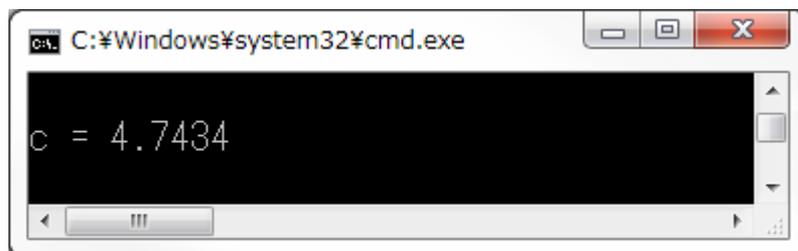
\*3 : 3.0の2.0乗 ( ) の中の値は必ず浮動小数点

## Step 3 修了課題

浮動小数点の変数  $a, b, c$  にそれぞれ下に示す値が代入されています。各設問で示すような出力になる場合に、以下のソースプログラムの `****` に入る適当な語句は何でしょうか。

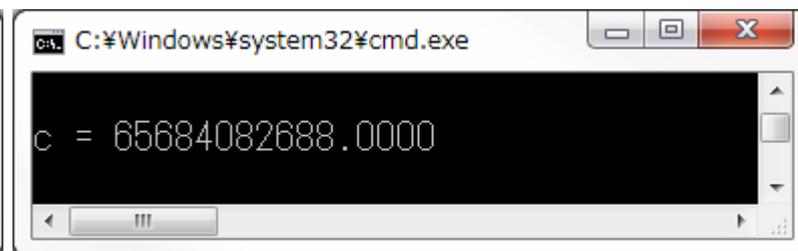
```
float a, b, c;  
a = 8.0; b = 22.5;  
c = ****;  
printf("c = %1.4f¥n", c);
```

出力画面①



```
C:\Windows\system32\cmd.exe  
c = 4.7434
```

出力画面②



```
C:\Windows\system32\cmd.exe  
c = 65684082688.0000
```

\* どちらの設問も、自分でプログラムを作成して、正解を確認してから回答するようにしましょう。



## Step 4

もしも、～だったら？

## 4-1 BMI値の計算プログラム

以下のソースプログラムを作成してビルドしてください。

```
int main()
{
    char buf[200];
    float weight, height, bmi;

    printf("¥nあなたの身長を入力してください(cm)¥t");
    gets_s(buf, sizeof(buf));
    height = atof(buf)/100; // 入力値をmに変換
    printf("¥nあなたの体重を入力してください(kg)¥t");
    gets_s(buf, sizeof(buf));
    weight = atof(buf);

    bmi = weight/(height*height); // BMI値を計算する

    printf("¥nあなたのBMI値は,%2.2fです。¥n", bmi);
    return 0;
}
```

## 4-2 if文による条件設定

```
if (条件式) {  
    処理  
}
```

## 4-3 BMI値の計算プログラム(続き)

4-1のソースプログラムに以下を追加してください。

```
printf("¥nあなたのBMI値は,%2.2fです。¥n", bmi);
```

```
// ここから追加分(注: return 0の前!)
```

```
if(bmi < 18.5){
```

```
    printf("あなたの体重は標準値より少ないです。¥n");
```

```
}
```

```
else if(bmi >= 18.5 && bmi <= 25.0){
```

```
    printf("あなたは標準体重です。¥n");
```

```
}
```

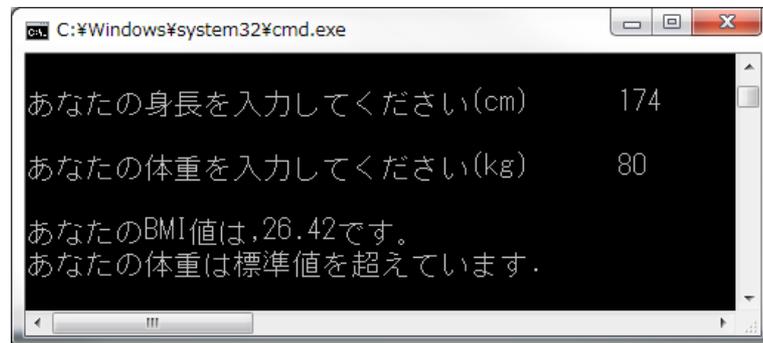
```
else {
```

```
    printf("あなたの体重は標準値を超えています。¥n");
```

```
}
```

```
}
```

出力例



```
C:\Windows\system32\cmd.exe
あなたの身長を入力してください(cm)      174
あなたの体重を入力してください(kg)      80
あなたのBMI値は,26.42です。
あなたの体重は標準値を超えています。
```

## 4-4 条件式に用いられる関係演算子

関係演算子	数学的表現	意味	プログラムでの表現	意味
==	=	等号	a==b	aとbは等しい
!=	≠	不等号	a!=b	aとbは等しくない
>=	≥	以上	a>=b	aはb以上である
<=	≤	以下	a<=b	aはb以下である
>	>	大なり	a>b	aはbよりも大きい
<	<	小なり	a<c	aはbよりも小さい

## 4-5 条件式に用いられる論理演算子

論理演算子	条件式	意味	プログラムでの表現例
AND	<code>a &amp;&amp; b</code>	a かつ b	<code>x &gt; 0 &amp;&amp; y == 0</code>
OR	<code>a    b</code>	a または b	<code>x == 1    x == 3</code>

# Cプログラミング上達のヒント！

## 変数の型を理解しましょう！

ソースプログラミング内で用いる変数は、intやfloat以外にも色々な型があります。目的に応じて使い分けるようにしましょう。

### 整数型 (integer type)

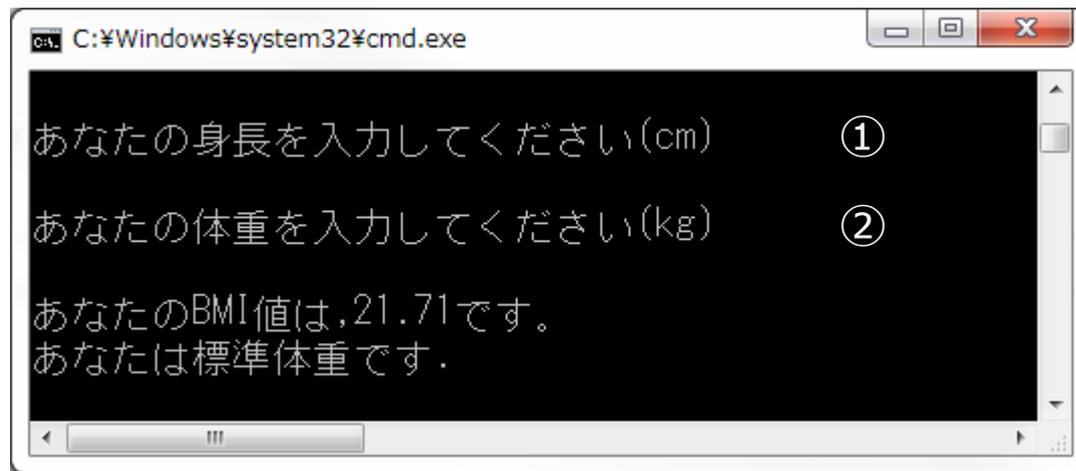
- char (1バイト) -128~127
- short int (2バイト) -32768~32768
- int (4バイト) -214783648~2147483647
- long int (4バイト) intと同じ

### 浮動小数点型 (float type)

- float (4バイト)  $3.4E-38 \sim 3.4E+38$
- double (8バイト)  $1.7E-308 \sim 1.7E+308$
- long double (8バイト)  $1.7E-308 \sim 1.7E+308$

## Step 4 修了課題

4-3で作成したプログラムを実行したら以下のような出力画面になりました。さて、このときに入力した①身長と②体重はいくらだったでしょうか？但し、入力した値は整数とします。



```
C:\Windows\system32\cmd.exe
あなたの身長を入力してください(cm) ①
あなたの体重を入力してください(kg) ②
あなたのBMI値は,21.71です。
あなたは標準体重です。
```

\* どちらの設問も、自分でプログラムを作成して、正解を確認してから回答するようにしましょう。



## Step 5

何度も繰り返してみましよう

## 5-1 星を書き出すプログラム

以下のソースプログラムを作成してビルドしてください。

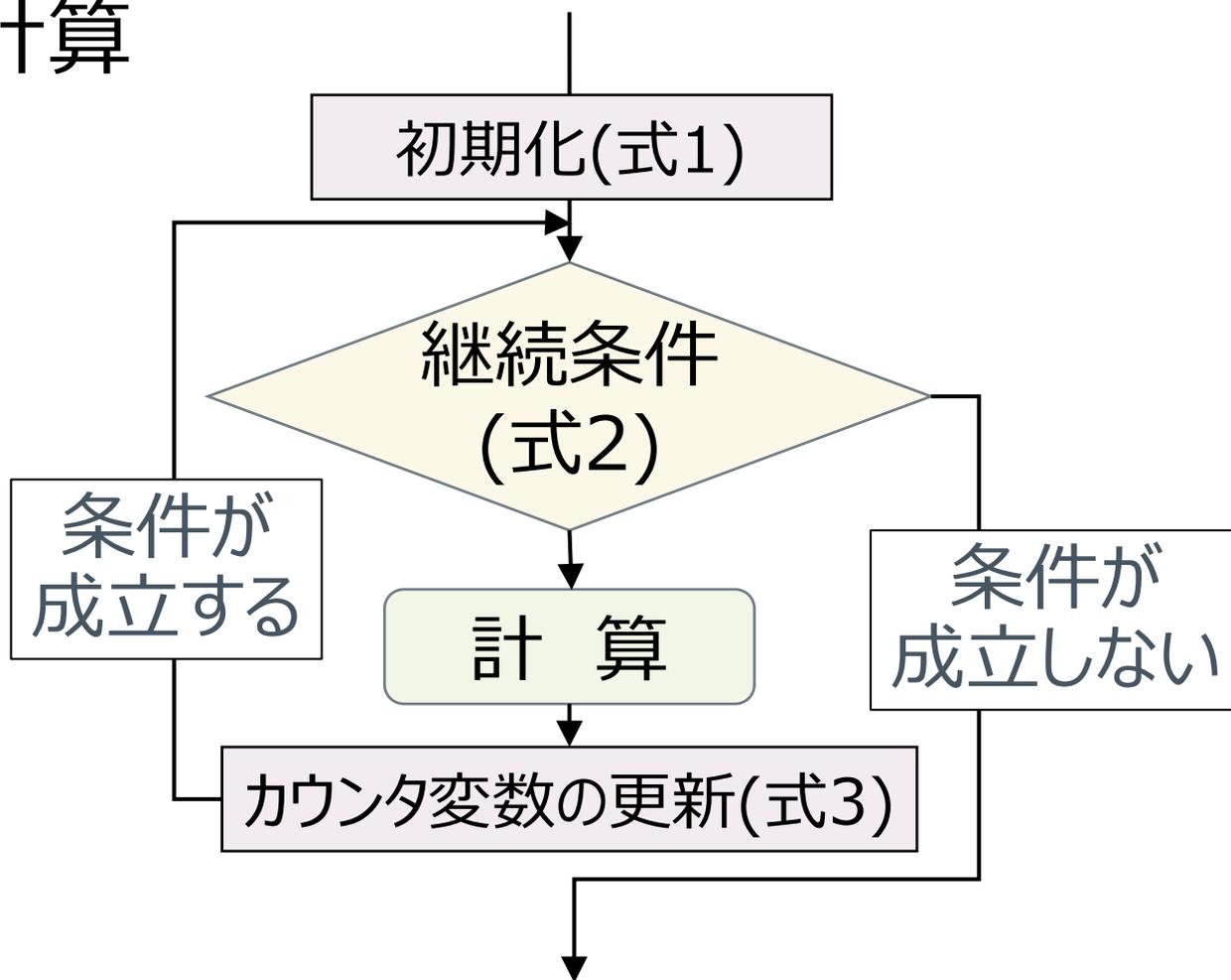
```
int main()
{
    char buf[200];
    int i, NumX;

    printf("¥n書き出したいXの数を入力してください)¥t");
    gets_s(buf, sizeof(buf));
    NumX = atoi(buf);

    for(i=0; i<NumX; i++){
        printf("X");
    }
    printf("¥n¥n"); // 最後に改行を追加
    return 0;
}
```

## 5-2 for文による繰り返し処理

```
for ( 式1; 式2; 式3) {  
    計算  
}
```



## 5-3 for文の意味

(A)

```
for(i=0; i<NumX; i++){  
    printf("X");  
}
```

(B)

(A) の後にfor文による繰り返し処理が始まる。

for文の中では、iは0から始まり、i++で1回繰り返すごとに1増加し、NumStarの数 - 1 になるまで、printf("X");の作業を繰り返す。繰り返し作業が終われば、プログラムは (B)へ進む。

# Cプログラミング上達のヒント！

## 「i++」って何？

Cプログラミングのソースプログラミングでは、ソースプログラムを簡略化するために、いくつかの省略形が用いられます。以下に代表的なものを示しますので、活用しましょう！

$i++ \rightarrow i = i + 1$  の意味。i以外の変数でも使用可

$x-- \rightarrow x = x - 1$  の意味

$y += 1 \rightarrow y++, y = y + 1$  と同じ意味

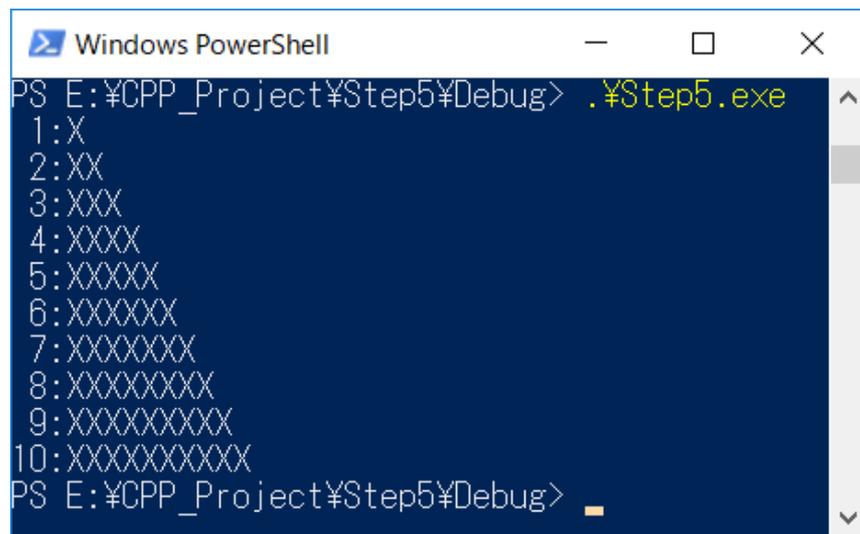
$y *= 2 \rightarrow y = y * 2$  の意味

## 5-4 二重の繰り返し

以下のソースプログラムを作成してビルドしてください。

```
int main()
{
    int i, j, NumX;
    NumX = 10;

    for(i=0; i<NumX; i++){
        printf("%2d:", i+1);
        for(j=0; j<i+1; j++){
            printf("X");
        }
        printf("¥n");
    }
    return 0;
}
```



```
Windows PowerShell
PS E:¥CPP_Project¥Step5¥Debug> .¥Step5.exe
1:X
2:XX
3:XXX
4:XXXX
5:XXXXX
6:XXXXXX
7:XXXXXXX
8:XXXXXXXX
9:XXXXXXXXX
10:XXXXXXXXXX
PS E:¥CPP_Project¥Step5¥Debug> _
```

出力例

## Step 5 修了課題

下に示したのは、100までの整数Xの階乗(X!)を計算するプログラムです。①,②,③,④に入る語句は何でしょう？

```
int main()
{
    char buf[100];
    int i, X, A;
    printf("¥n100までの整数を入力して下さい。 ¥t");
    gets_s(buf, sizeof(buf));
    X = atof(buf); A = 1;
    for ( i = ① ; i ② ; i ③ ){
        A ④ i;
    }
    printf("%dの階乗は%dです。 ¥n", X, A);
    return 0;
}
```



## Step 6

### 画像データの複製

## 6-0 事前準備（重要！）

今回のStepから、ソースプログラムの1行目に以下の1行を必ず追記してください。

```
#define _CRT_SECURE_NO_WARNINGS
```

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <string.h>
5  #include <math.h>
6
7  int main()
8  {
9      unsigned char* InImage;    //入力画像データの配列の宣言
10     unsigned char* OutImage;   //出力画像データの配列宣言
```

この1行を追加

## 6-1 画像を複製するプログラム①

以下の256x256の画像を複製するプログラムのソースプログラムを作成してビルドしてください。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

int main()
{
    unsigned char *InImage;           //入力画像データの配列の宣言
    unsigned char *OutImage;         //出力画像データの配列宣言
    char InFile[100], OutFile[100]; //入出力画像ファイル名の宣言
    int i, Xsize, Ysize;
    FILE *fp;                         // fopen関数で必要な入力バッファの宣言
    printf("¥n入力する画像のファイル名を入力してください)¥t");
    gets_s(InFile, sizeof(InFile));
    printf("¥n出力する画像のファイル名を入力してください)¥t");
    gets_s(OutFile, sizeof(OutFile));
```

## 6-1 画像を複製するプログラム②

(ここから)

```
Xsize = 256; Ysize = 256;
InImage = new unsigned char[Xsize*Ysize];
OutImage = new unsigned char[Xsize*Ysize];
// 配列InImage, OutImageのメモリ領域を動的に確保

if ((fp = fopen(InFile, "rb")) == NULL){
    printf("Could not open filename %s ¥n", InFile);
    exit(0);
}
fread(InImage, sizeof(unsigned char), Xsize*Ysize, fp);
fclose(fp);

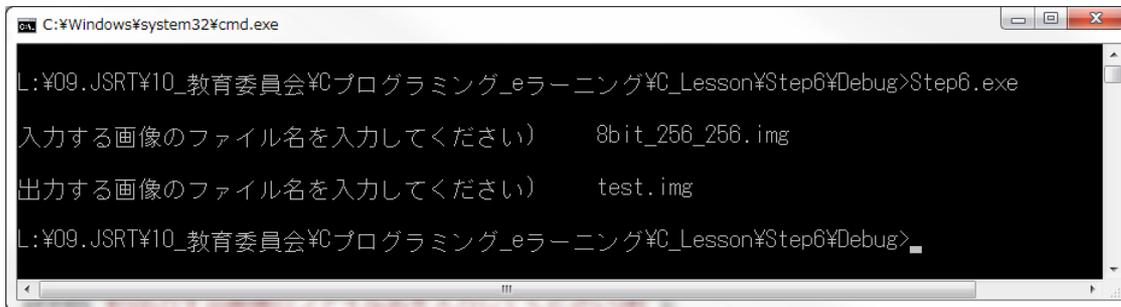
for (i = 0; i<Xsize*Ysize; i++){
    OutImage[i] = InImage[i]; //InImageの中身をOutImageに複製
}
```

(次へ続く)

# 6-1 画像を複製するプログラム③

(ここから)

```
if ((fp = fopen(OutFile, "wb")) == NULL){  
    printf("Could not open filename %s ¥n", OutFile);  
    exit(0);  
}  
fwrite(OutImage, sizeof(unsigned char), Xsize*Ysize, fp);  
fclose(fp);  
  
delete[] InImage;  
delete[] OutImage;  
return 0;  
}
```



```
C:\Windows\system32\cmd.exe  
L:\¥09.JSRT¥10_教育委員会¥Cプログラミング_エラーニング¥C_Lesson¥Step6¥Debug>Step6.exe  
入力する画像のファイル名を入力してください) 8bit_256_256.img  
出力する画像のファイル名を入力してください) test.img  
L:\¥09.JSRT¥10_教育委員会¥Cプログラミング_エラーニング¥C_Lesson¥Step6¥Debug>_
```

出力例と複製された画像



## 6-2 配列と動的・静的メモリの確保

### 1) 配列の概念

配列とは、複数の変数に番号を付けて一つにまとめたもの

```
int i, j, k;
```

i	j	k
4バイト	4バイト	4バイト

```
int ijk[3];
```

ijk[0]	ijk[1]	ijk[2]
4バイト	4バイト	4バイト

## 6-2 配列と動的・静的メモリの確保

### 2) 静的メモリと動的メモリ

静的メモリ：特定の配列のために、プログラム起動時から終了時まで固定したメモリ領域をコンピュータ内に確保する。

例：`int hairetu[500];`

動的メモリ：特定の配列のために、プログラム動作中に必要に応じてメモリを確保・開放する。

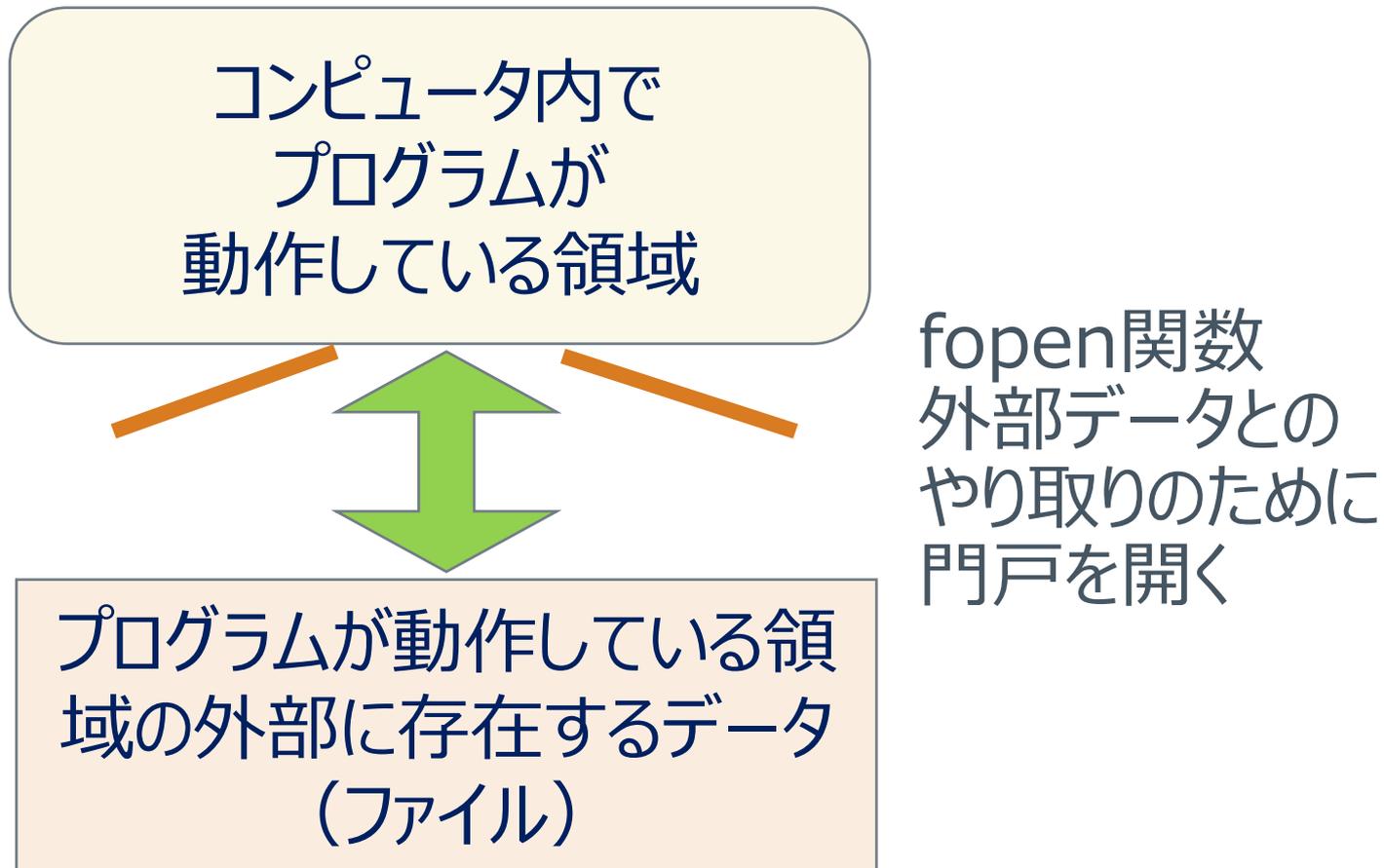
例：`int *hairetu;`

確保 `hairetu = new int[500];`

開放 `delete [] hairetu;`

## 6-3 ファイルの入力・出力

### 1) fopen関数



## 6-3 ファイルの入力・出力

### 2) fclose関数

コンピュータ内で  
プログラムが  
動作している領域



プログラムが動作している領  
域の外部に存在するデータ  
(ファイル)

fclose関数  
外部データとの  
やり取りのために  
開けた門戸を閉  
じる

## 6-4 fread関数とfwrite関数

fread関数：画像ファイルのデータ型とサイズを指定して読み込む

```
fread(入力データ配列,sizeof(データ型),サイズ,fp);
```

fwrite関数：画像ファイルのデータ型とサイズを指定して書き出す

```
fwrite(出力データ配列,sizeof(データ型),サイズ,fp);
```

FILE \*fp;で宣言している場合

## Step 6 修了課題

6-1 で作成したプログラムを512×512の画像を複製するように変更して、そのプログラムで作成した以下の画像の複製をアップロードしてください。



JPCLN001\_512.raw



# Step 7

## 画像の階調処理

## 7-1 画像の階調処理のプログラム①

Step7では、6-1のプログラムの②の一部を変更して新しいプログラムを作成します(①と③は変更なし)。

6-1 ②の  の部分を次ページからのソースプログラムと置き換えます。

```
Xsize = 256;   Ysize = 256;
InImage = new unsigned char[Xsize*Ysize];
OutImage = new unsigned char[Xsize*Ysize];
// 配列InImage, OutImageのメモリ領域を動的に確保

if ((fp = fopen(InFile, "rb")) == NULL){
    printf("Could not open filename %s ¥n", InFile);
    exit(0);
}
fread(InImage, sizeof(unsigned char), Xsize*Ysize, fp);
fclose(fp);

for (i = 0; i<Xsize*Ysize; i++){
    OutImage[i] = InImage[i]; //InImageの中身をOutImageに複製
}
```

## 7-1 画像の階調処理のプログラム②

```
float ganma, center;
```

```
char buf[100];
```

```
float pv;
```

```
printf("¥nコントラスト(-2.0~2.0)を入力してください¥t");
```

```
gets_s(buf, sizeof(buf));
```

```
ganma = atof(buf);
```

```
printf("¥n回転中心(0~255)を入力してください¥t");
```

```
gets_s(buf, sizeof(buf));
```

```
center = atof(buf);
```

```
if (ganma < -2.0 || ganma > 2.0 || center > 255 || center < 0){
```

```
    printf("入力されたデータに間違いがあります. ¥n");
```

```
    exit(0);
```

```
}
```

## 7-1 画像の階調処理のプログラム③

```
for (i = 0; i < Xsize*Ysize; i++){  
    pv = (float)(InImage[i] - center)* gamma + center;  
    if (pv < 0)    pv = 0;  
    if (pv > 255) pv = 255;  
    OutImage[i] = (unsigned char)pv;  
}
```

プログラムの出力例（回転中心はどちらも128）



オリジナル画像

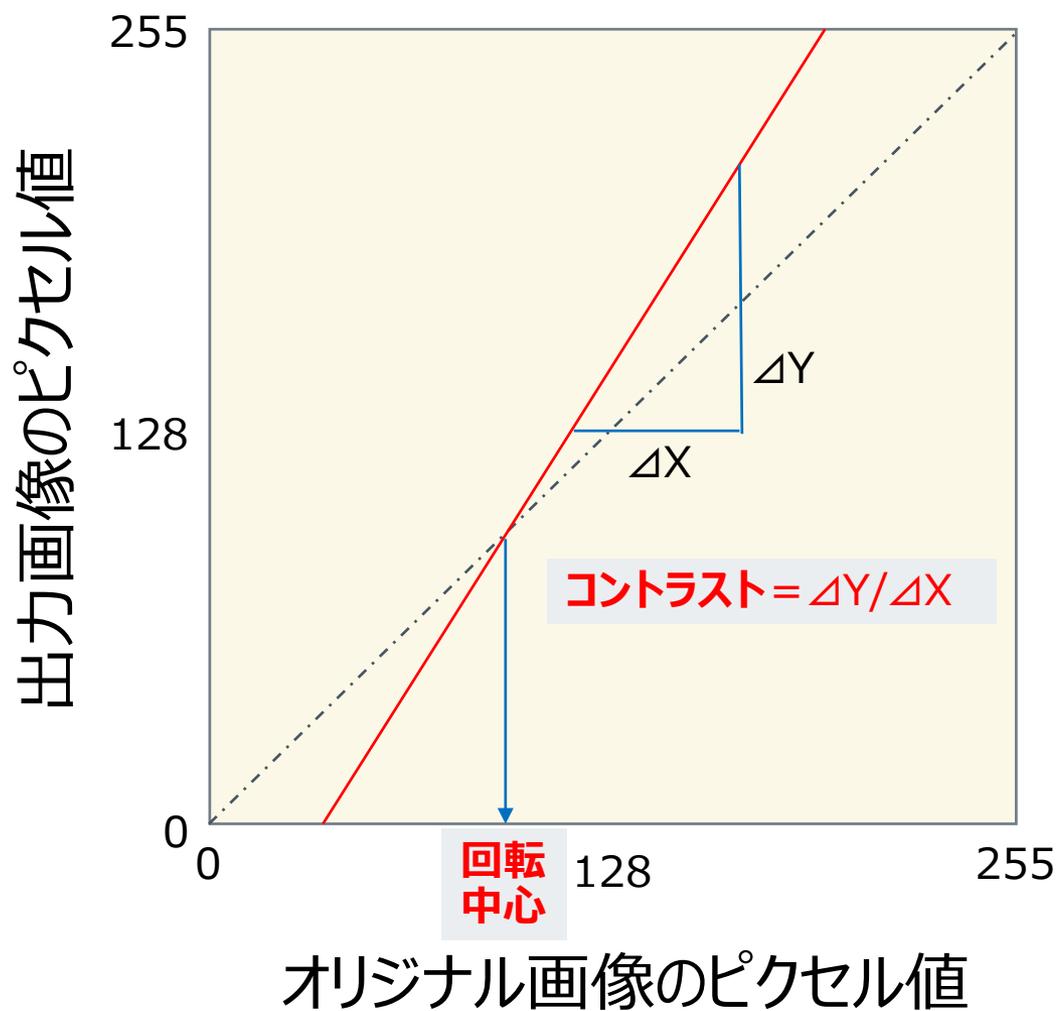


コントラスト：-1.0



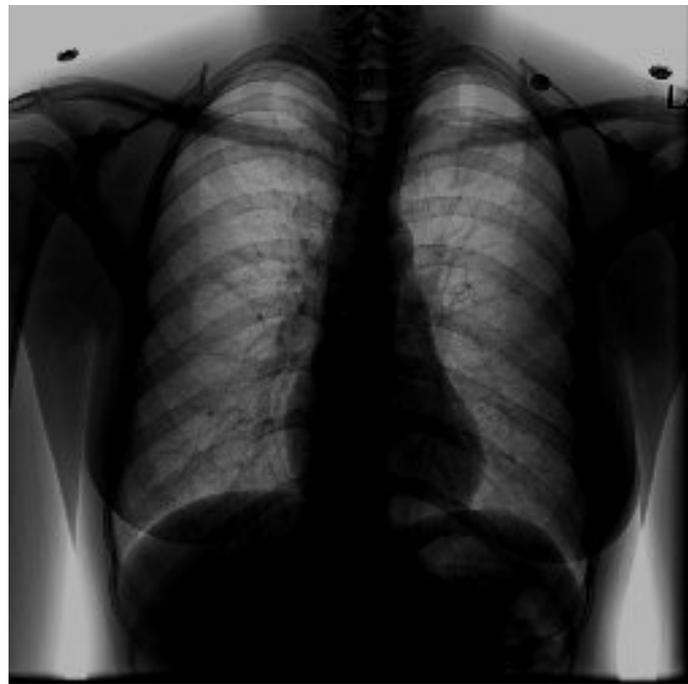
コントラスト：1.5

## 7-2 線形変換による階調処理



## Step 7 修了課題

7-1 で作成したプログラムを用いて、  
8bit\_256\_256.imgのテスト画像から、コントラスト  
が $-0.9$ 、回転中心が100の画像を作成してアップ  
ロードしてください。



回答例



## Step 8

### 画像の拡大と縮小

## 8-1 画像の拡大縮小のプログラム①

以下のソースプログラムを入力し、ビルドしてください。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

int main()
{
    unsigned char *InImage, *OutImage;
    char buf[100], InFile[100], OutFile[100];
    int i, InSize, OutSize;

    printf("¥n入力する画像のファイル名を入力してください)¥t");
    gets_s(InFile, sizeof(InFile));
    printf("¥n入力する画像の一辺の長さを入力してください)¥t");
    gets_s(buf, sizeof(buf));
    InSize = atoi(buf);
```

## 8-1 画像の拡大縮小のプログラム②

```
printf("¥n出力する画像のファイル名を入力してください)¥t");
gets_s(OutFile, sizeof(OutFile));
printf("¥n出力する画像の一辺の長さを入力してください)¥t");
gets_s(buf, sizeof(buf));
OutSize= atoi(buf);

InImage = new unsigned char[InSize*InSize];
OutImage = new unsigned char[OutSize*OutSize];

double mag;
FILE *fp;           // fopen関数で必要な入力バッファの宣言
if ((fp = fopen(InFile, "rb")) == NULL){
    printf("Could not open filename %s ¥n", InFile);
    exit(0);
}
fread(InImage, sizeof(unsigned char), InSize*InSize, fp);
fclose(fp);
mag = (double)OutSize / (double)InSize;
```

## 8-1 画像の拡大縮小のプログラム③

```
for (i = 0; i<OutSize*OutSize; i++) OutImage[i] = 0;
```

```
float origX, origY, dx, dy, vA, vB, vL;
```

```
int x, y, ftY, ftX;
```

```
float px11, px12, px21, px22;
```

```
for (y = 0; y<OutSize; y++){
```

```
    origY = (float)y/ mag;
```

```
    ftY = (int)origY;
```

```
    dy = origY - (float)ftY;
```

```
    if ((ftY>0) && (ftY<InSize - 1)){
```

```
        for (x = 0; x<OutSize; x++){
```

```
            origX =(float)x/ mag;
```

```
            ftX = (int)origX;
```

```
            dx = origX - (float)ftX;
```

```
            if ((ftX > 0) && (ftX < InSize - 1)){
```

```
                px11 = (float)InImage[ftY*InSize + ftX];
```

```
                px21 = (float)InImage[ftY*InSize + ftX + 1];
```

## 8-1 画像の拡大縮小のプログラム④

```
        px12 = (float)InImage[(ftY + 1)*InSize + ftX];
        px22 = (float)InImage[(ftY + 1)*InSize + ftX + 1];
        vA = (1.0 - dx)*px11 + dx*px21;
        vB = (1.0 - dx)*px12 + dx*px22;
        vL = (1.0 - dy)*vA + dy*vB;
        OutImage[y*OutSize + x] = (unsigned char)vL;
    }
}
}

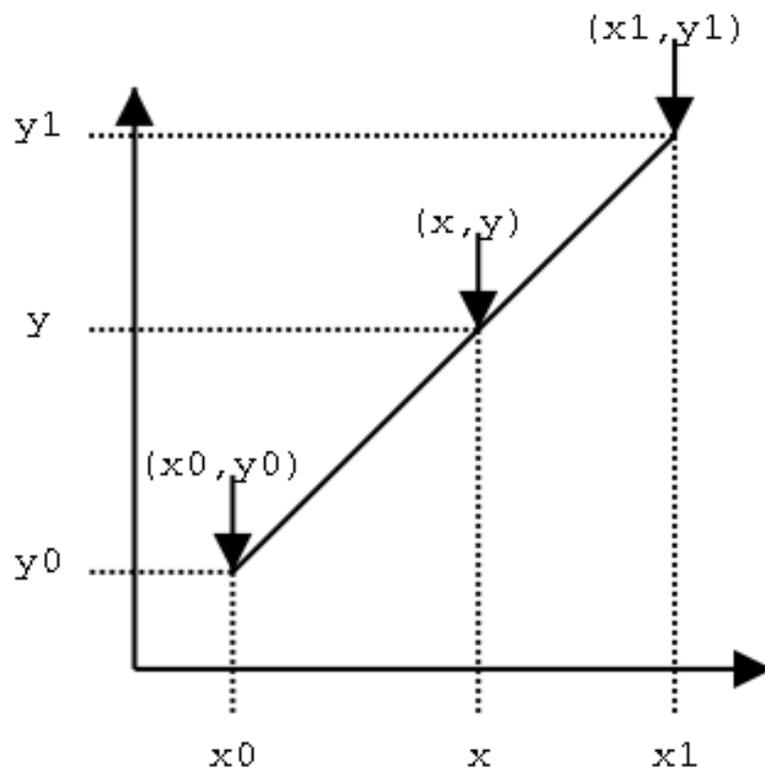
if ((fp = fopen(OutFile, "wb")) == NULL){
    printf("Could not open filename %s ¥n", OutFile);
    exit(0);
}
fwrite(OutImage, sizeof(unsigned char), OutSize*OutSize, fp);
fclose(fp);
delete[] InImage;
delete[] OutImage;
return 0;
}
```

## 8-2 線形補間による画像の拡大・縮小

$$\alpha = \frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0}$$

$$y = (1 - \alpha)y_0 + \alpha y_1$$

$$y = y_0 + \alpha(y_1 - y_0)$$



## Step 8 修了課題

8-1 で作成したプログラムを用いて、  
8bit\_256\_256.imgのテスト画像からマトリクスサイズが  
300×300の画像を作成してアップロードしてください。



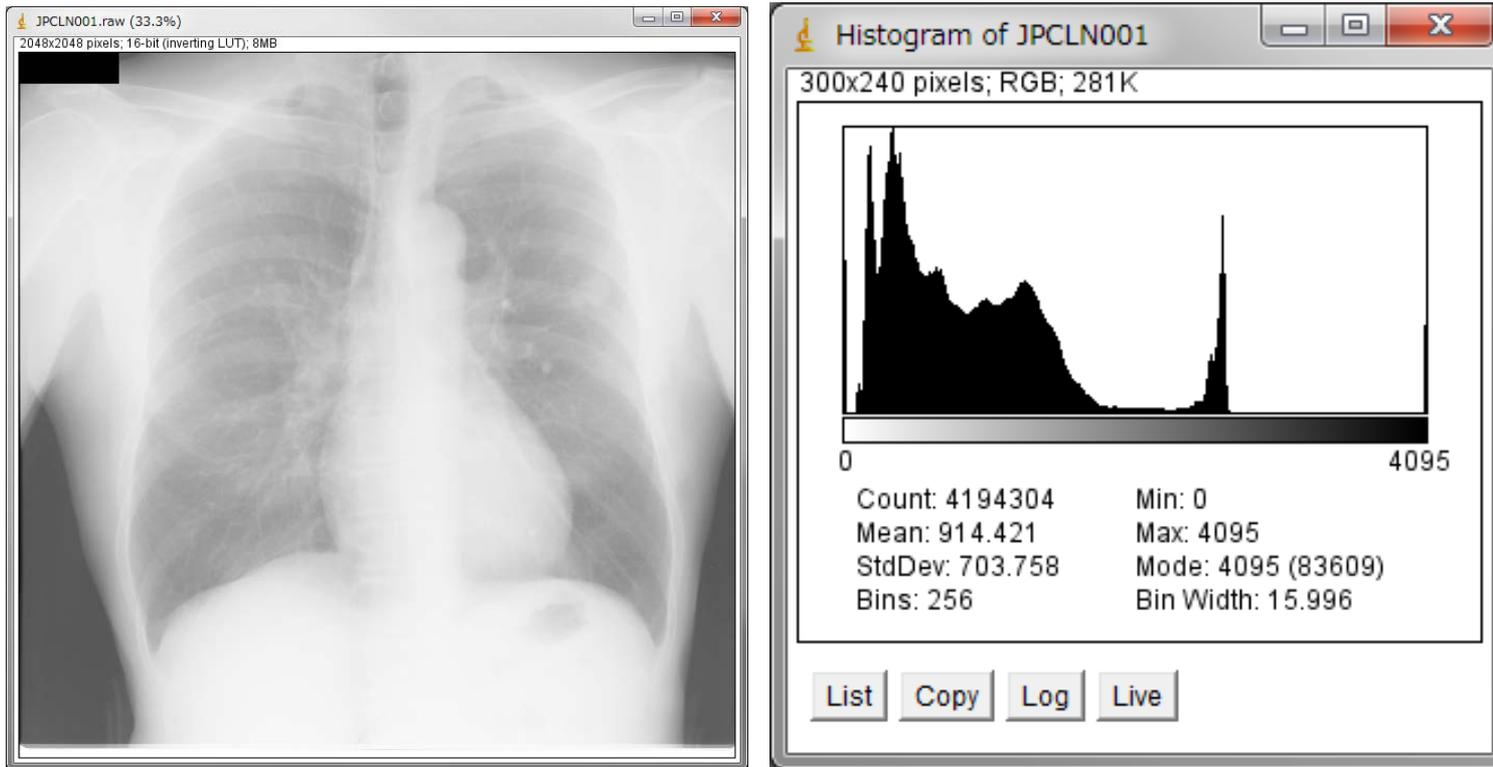
回答例



## Step 9

### 画像のヒストグラム

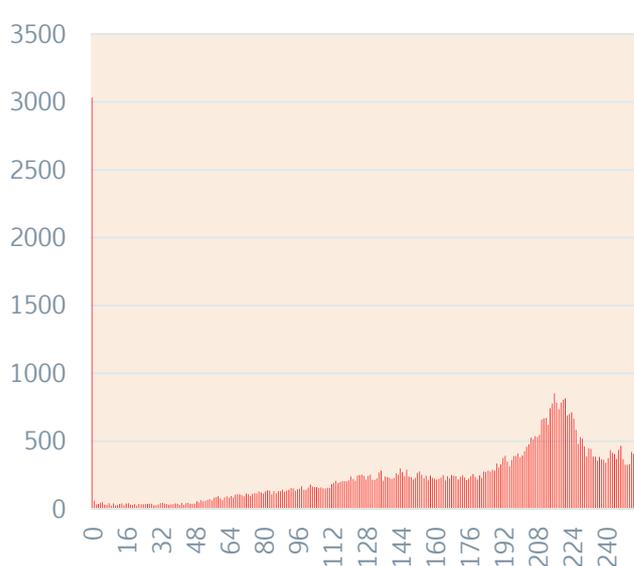
## 9-1 画像のヒストグラムとは？



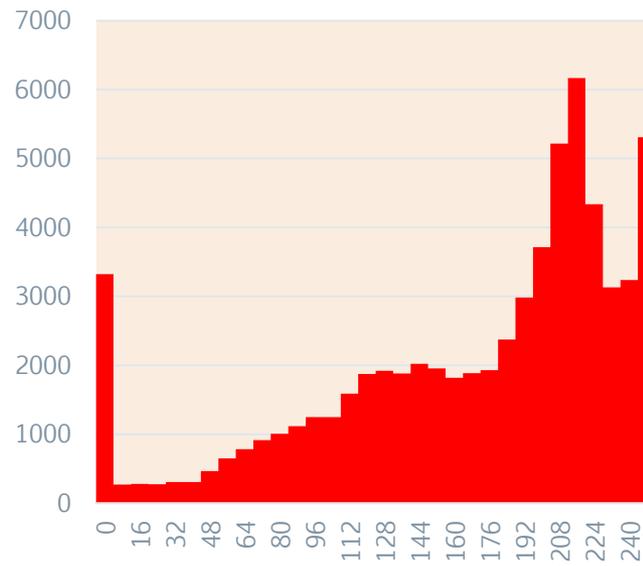
画像に含まれる画素の値の頻度分布図

## 9-2 画像のヒストグラムとBinサイズ

ヒストグラムのスケールは画像の階調数に等しい。ピクセル値のスケール一つ一つの頻度には変動が多いので、通常、ヒストグラムを表示する場合は、いくつかの値ごとにまとめた数値で表現する。このまとめたピクセル値の数のことをBinという。



Bin = 1の時



Bin = 8の時

## 9-3 ヒストグラム作成のプログラム①

以下のソースプログラムを入力し，ビルドしてください。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

int main()
{
    unsigned char *InImage;
    unsigned char *OutImage; // Step10で使用
    char buf[100], InFile[100];
    char OutFile[100]; // OutFileもStep10で使用
    int i, j, pv, InSize, *Hist; // ヒストグラム作成用の配列
    int gs = 256; // 8bit画像なので256階調
    FILE *fp; // fopen関数で必要な入力バッファの宣言
```

## 9-3 ヒストグラム作成のプログラム②

```
printf("¥n入力する画像のファイル名を入力してください)¥t");
gets_s(InFile, sizeof(InFile));
printf("¥n入力する画像の一辺の長さを入力してください)¥t");
gets_s(buf, sizeof(buf));
InSize = atoi(buf);

InImage = new unsigned char[InSize*InSize];
Hist = new int[gs]; // 256階調分(0 - 255)の配列を確保

for (i = 0; i < gs; i++) Hist[i] = 0; // 配列Histの初期化

if ((fp = fopen(InFile, "rb")) == NULL){
    printf("Could not open filename %s ¥n", InFile);
    exit(0);
}
fread(InImage, sizeof(unsigned char), InSize*InSize, fp);
fclose(fp);
```

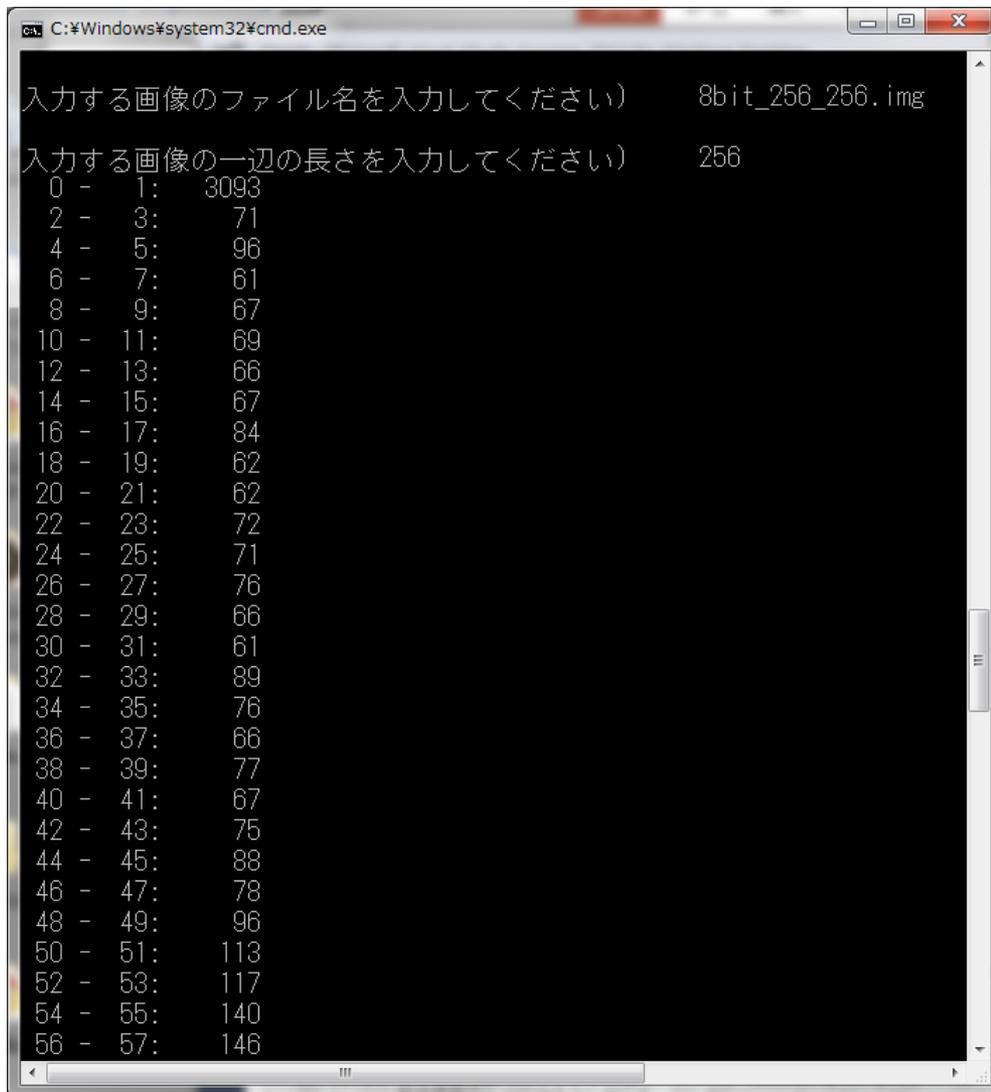
## 9-3 ヒストグラム作成のプログラム③

```
for (i = 0; i < InSize*InSize; i++){
    pv = (int)InImage[i];
    Hist[pv]++; // 配列の中のpvの値の数を1つ加算
}

int bin = 2; // binサイズの決定
int sum = 0; // binの内部の小計の計算用

for (i = 0; i < gs / bin; i++){
    printf("%3d - %3d: ", i*bin, i*bin + bin - 1);
    sum = 0;
    for (j = 0; j < bin; j++) sum += Hist[i*bin + j];
    printf("%6d¥n", sum);
}
delete[]Hist;
delete[]InImage;
return 0;
}
```

## 9-4 ヒストグラムの出力例

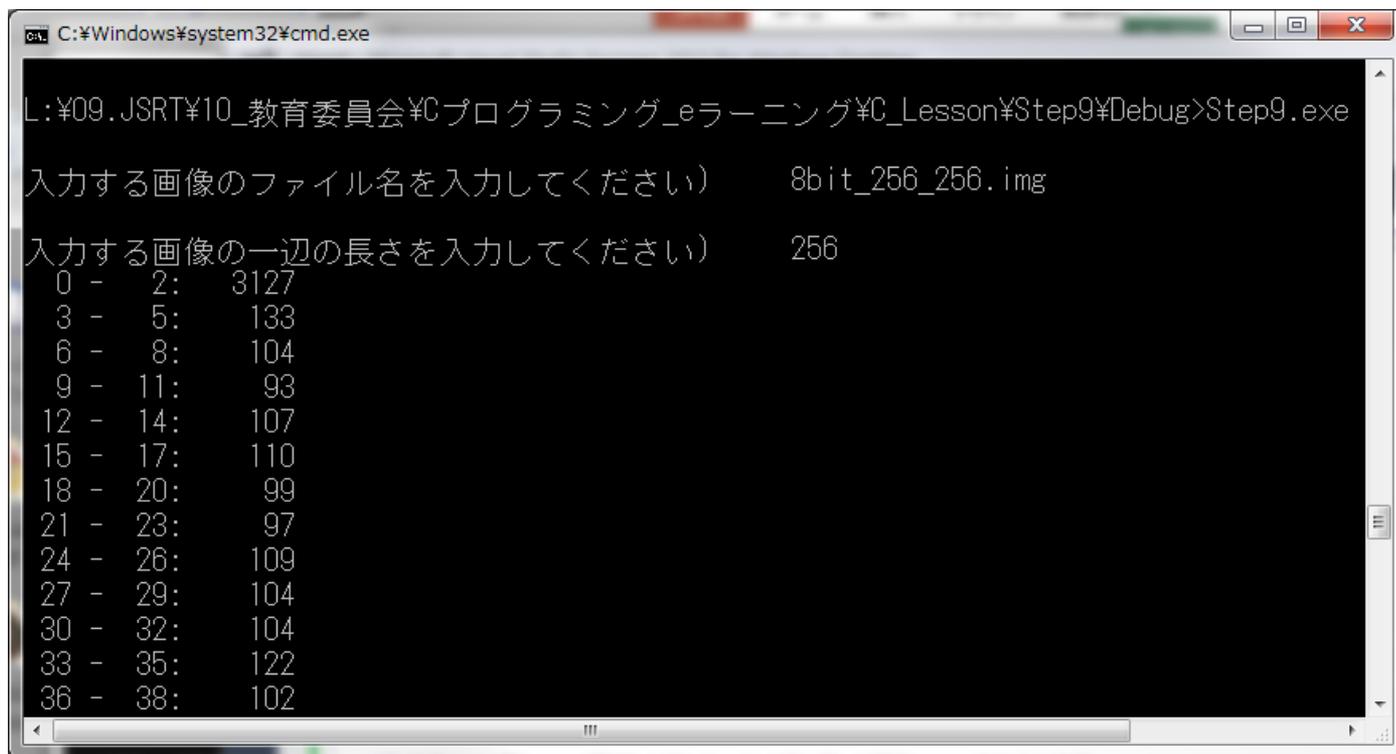


The screenshot shows a Windows command prompt window with the following text:

```
C:\Windows\system32\cmd.exe
入力する画像のファイル名を入力してください) 8bit_256_256.img
入力する画像の一辺の長さを入力してください) 256
0 - 1: 3093
2 - 3: 71
4 - 5: 96
6 - 7: 61
8 - 9: 67
10 - 11: 69
12 - 13: 66
14 - 15: 67
16 - 17: 84
18 - 19: 62
20 - 21: 62
22 - 23: 72
24 - 25: 71
26 - 27: 76
28 - 29: 66
30 - 31: 61
32 - 33: 89
34 - 35: 76
36 - 37: 66
38 - 39: 77
40 - 41: 67
42 - 43: 75
44 - 45: 88
46 - 47: 78
48 - 49: 96
50 - 51: 113
52 - 53: 117
54 - 55: 140
56 - 57: 146
```

## Step 9 修了課題

9-3 で作成したプログラムのbinサイズを3に変更して、8bit\_256\_256.imgのテスト画像のヒストグラムを求め、ピクセル値が210-212の範囲のピクセルの数を回答してください。



```
C:\Windows\system32\cmd.exe
L:\09.JSRT\10_教育委員会\Cプログラミング_eラーニング\C_Lesson\Step9\Debug>Step9.exe
入力する画像のファイル名を入力してください) 8bit_256_256.img
入力する画像の一边の長さを入力してください) 256
0 - 2: 3127
3 - 5: 133
6 - 8: 104
9 - 11: 93
12 - 14: 107
15 - 17: 110
18 - 20: 99
21 - 23: 97
24 - 26: 109
27 - 29: 104
30 - 32: 104
33 - 35: 122
36 - 38: 102
```

出力例

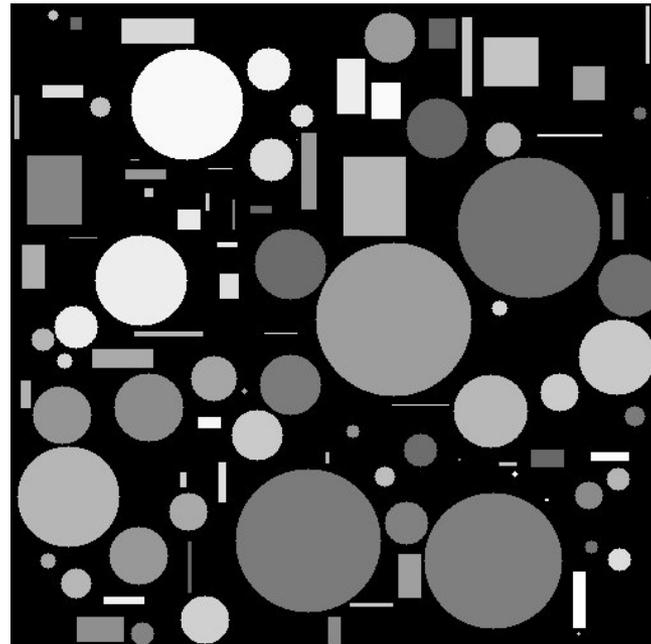


# Step 10

ウォーリーを探せ！

# 10-1 最後の課題「ウォーリーを探せ！」

初級講座の最後の課題です。下に示した課題画像 (WhereIsWally.img) を読み込んで、この画像全体のピクセル値の中央値 (メディアン) に最も近い図形を探し出して、それ以外を黒く塗りつぶします。



# 10-1 Where's Wallyのプログラム①

π

Step10 では, 9-1のヒストグラムのプログラムの③の一部を変更して新しいプログラムを作成します(①と②は変更なし).

```
for (i = 0; i < InSize*InSize; i++){  
    pv = (int)InImage[i];  
    Hist[pv]++; // 配列の中のpvの値の数を1つ加算
```

```
}
```

ここから下の部分を次ページ以降のソースと入れ替えます.

```
int bin = 2; // binサイズの決定  
int sum = 0; // binの内部の小計の計算用  
for (i = 0; i < gs / bin; i++){  
    printf("%3d - %3d: ", i*bin, i*bin + bin - 1);  
    sum = 0;  
    for (j = 0; j < bin; j++) sum += Hist[i*bin + j];  
    printf("%6d¥n", sum);
```

```
}
```

```
delete[] Hist;  
delete[] InImage;  
return 0;
```

```
}
```

# 10-1 Where's Wallyのプログラム②

```
int bin = 1; // binサイズの決定
int sum = 0;
int med = 0;

printf("¥n出力する画像のファイル名を入力してください)¥t");
gets_s(OutFile, sizeof(OutFile));

OutImage = new unsigned char[InSize*InSize];

for (i = 0; i < InSize*InSize; i++) OutImage[i] = 0;

for (i = gs-1; i >0; i--){
    sum += Hist[i];
    if ((float)sum <= (float)(InSize*InSize )*0.50){
        if(Hist[i] != 0) med = i;
    }
}

printf("%sのメディアンは%dです. ¥n", InFile, med);
```

# 10-1 Where's Wallyのプログラム③

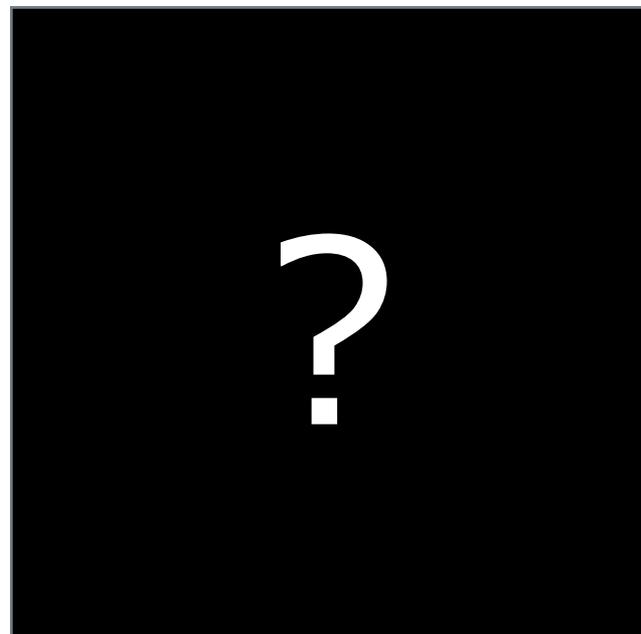
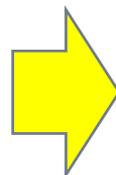
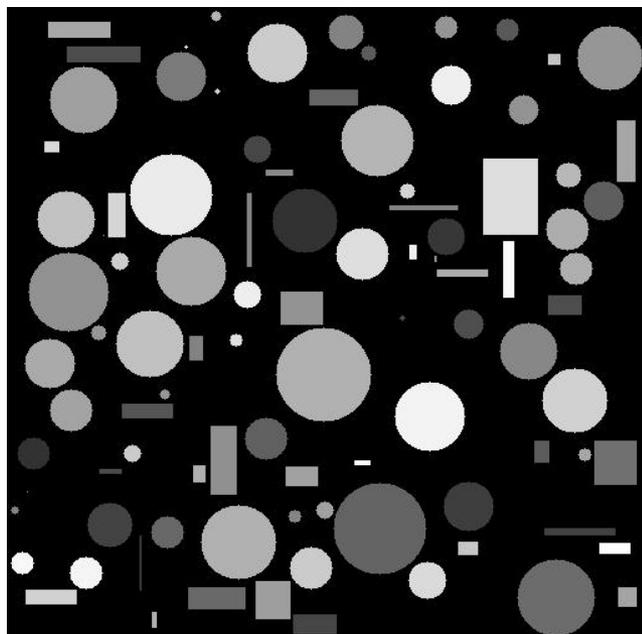
```
for (i = 0; i < InSize*InSize; i++){
    pv = InImage[i];
    if (pv != med) OutImage[i] = 0;
    else OutImage[i] = InImage[i];
}

if ((fp = fopen(OutFile, "wb")) == NULL){
    printf("Could not open filename %s ¥n", OutFile);
    exit(0);
}
fwrite(OutImage, sizeof(unsigned char), InSize*InSize, fp);
fclose(fp);

delete[]Hist;
delete[]InImage;
delete[]OutImage;
return 0;
}
```

## Step 10 最終課題

10-1 で作成したプログラムで出力された図形 (Wally) だけが残った画像をアップロードしてください。



どうもお疲れ様でした！

# おまけ (特別課題)

\* 一つぐらいは自分の力でプログラムを考えてみたいという方のために・・・

10-1 で作成したプログラムで残った画像全体のピクセル値のメディアンに最も近い図形について，その重心の座標とピクセル値はいくらでしょうか？ 重心座標を計算するプログラムの考え方は以下に示す通りです。

y\x	0	1	2	3	4	5
0						
1			1			
2		2	3	4		
3	5	6	7	8	9	10
4		11	12	13	14	15
5				16	17	18
6		19	20	21	22	23
7				24	25	26

重心(3,4)

重心(x,y): 候補領域内のピクセルの X, Yそれぞれの座標の  
平均値

$$x = \frac{1}{n} \sum x_i$$

$$x = \frac{0*1 + 1*4 + 2*5 + 3*6 + 4*5 + 5*5}{26}$$

$$= \frac{77}{26} \approx 3$$

$$y = \frac{1}{n} \sum y_i = \frac{111}{26} \approx 4$$



最後に

この体験講座のLimitation

# この初級講座では勉強できなかったこと

この初級講座は、C言語プログラミングがまったく初めてという方を対象として、理論を学ぶことよりも実際にプログラムを書いて、実行することを主体において実施されています。そのため、以下に示すような画像処理プログラミングに必要な項目はあえて含めませんでした。

- コマンドライン入力に関すること
- ヘッド情報に関すること
- サブルーチンに関すること
- メモリの確保に関すること etc..

今回の初級講座で、C言語プログラミングに興味を持たれた方は、さらなる勉強を始められることを強く推奨します。

画像部会の画像処理プログラミングセミナーで  
お待ちしております。