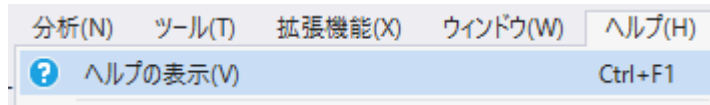
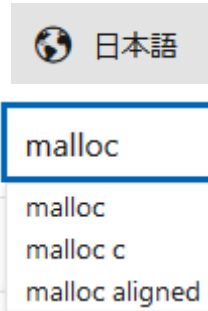


ヘルプを見てみよう！

- 関数などで分からないことがあるときは、基本的にはヘルプを見る
 - 最近では、付属のヘルプを見るより、Google検索で調べたほうが早い
 - 一部の情報は日本語版がなく、英語版しかないものもある
 - OpenGLなどの外部のライブラリに関する情報は、基本的にはヘルプには含まれていない
- [ヘルプ]-[ヘルプの表示]



- 左下が[English (United States)]などになっている場合はクリックして[日本語]を選ぶ
- 右上の[検索]をクリック **検索** 🔍
- 検索したい関数(この例の場合は[malloc])を入力してEnterキーを押す



- 検索結果から知りたい情報をクリック(この例の場合は[malloc])

C++ における "malloc" の 133 件の結果

🔊 RS

または [Microsoft Docs のすべての結果を表示](#)

すべて (133) ドキュメント (133) Learn (0) リファレンス (0)

malloc

[/cpp/c-runtime-library/reference/malloc](#)

When the application is linked with a debug version of the C run-time libraries, `malloc` resolves to `_malloc_dbg`. ... Example See also `Memory Allocation` `calloc` `free` `realloc` `_aligned_malloc`

- 知りたい情報(この例の場合はmalloc関数)が出てくる
- 例えばmalloc関数のヘルプをみた場合
- 「メモリブロックを割り当てる」関数ということが分かる
- 割り当てるバイト数を引数sizeで指定することが分かる
- 戻り値は割り当てたメモリへのポインタということが分かる
- stdlib hかmalloc hをincludeすればいいことが分かる

malloc

2016/11/04 • 👤 🌟 🌟 🌟 🌟 +4

メモリブロックを割り当てます。

構文

```
C
void *malloc(
    size_t size
);
```

パラメーター

`size`
割り当てるバイト数。

戻り値

`malloc`は、割り当てられた領域への void ポインタを返します。

- 日本語訳が下手すぎて、IT技術の勉強には英語の文章を読んで知識を習得したほうがいいことが分かる

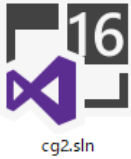
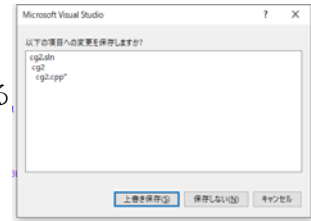
日本語版を見ると意味不明 → **Malloc**関数は、サイズが少なくともバイトのメモリブロックを割り当てます。

英語版に正しい情報が載っている → **The malloc function allocates a memory block of at least size bytes.**

正しくは「malloc関数は少なくともsizeバイトのメモリブロックを割り当てます」

終了

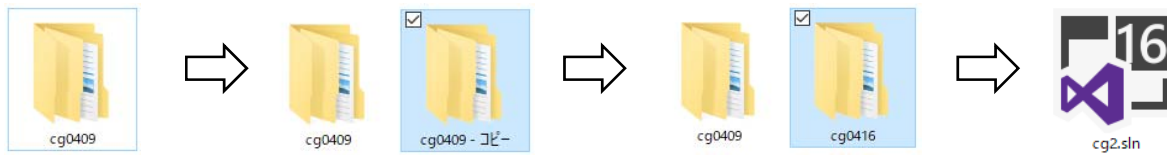
- 1 終了するときには必ず保存しよう
- 2 ビルド時は自動的に保存される
- 3 保存せずに終了しようとするとき[保存しますか?]--と聞かれるので[はい]--と答える
- 4 もう一度開くときは.slnファイルをダブルクリック



cg2.sln

Visual C++のプロジェクトを使い回す方法

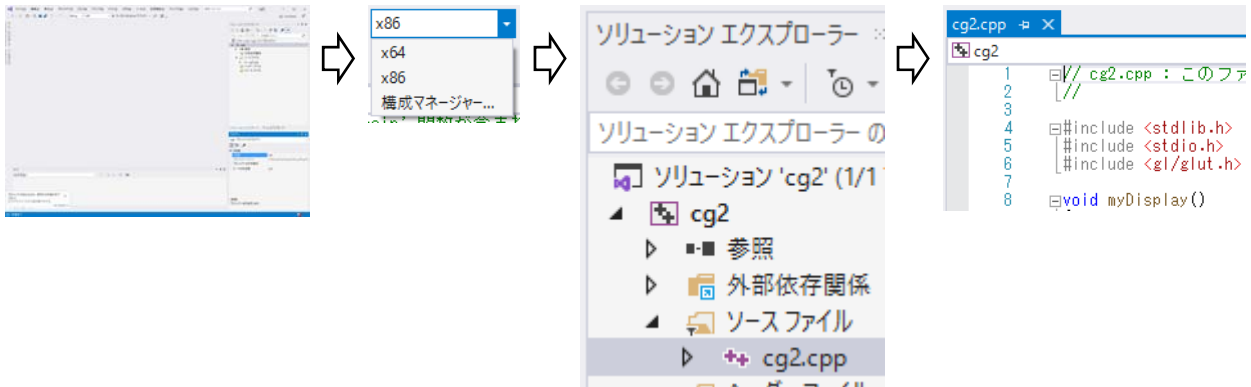
- 1 前回のフォルダを右クリックして[コピー]
- 2 エクスプローラの空白部分を右クリックして[貼り付け]
- 3 コピーされたフォルダを右クリックして[名前の変更]
- 4 そのフォルダの中にあるslnファイルをダブルクリック



- 1 フォルダごとコピーするのであって、一部のファイルだけをコピーするわけではない
- 2 フォルダ名を変更するのであって、cppファイルやslnファイルなどファイル名を変更するわけではない
- 3 slnファイルをダブルクリックするのであって、cppファイルをダブルクリックするわけではない

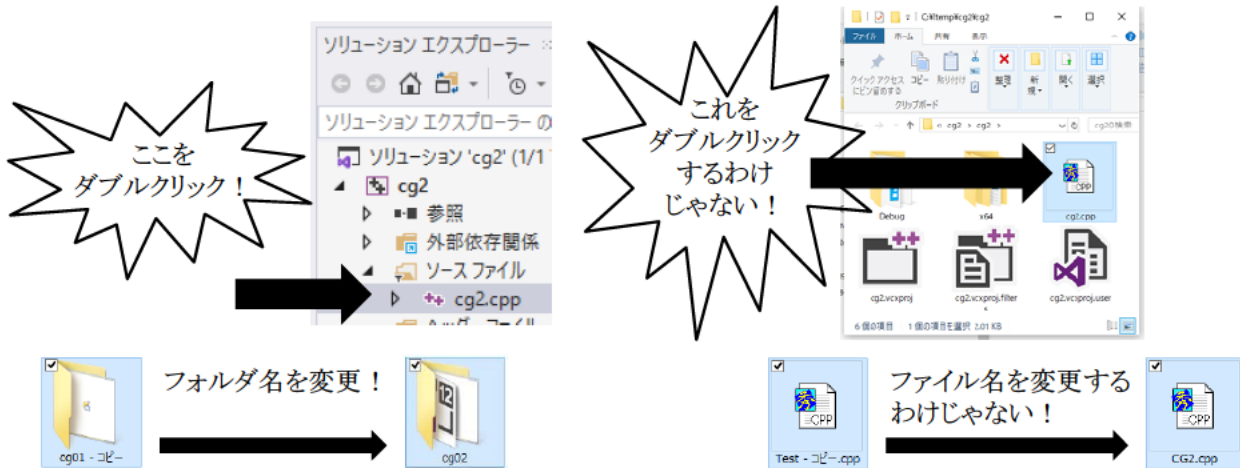
cppファイルを開く

- 1 Visual Studioが立ち上がる
- 2 [ソリューションプラットフォーム]の[x64]を[x86]にする
- 3 右にある[ソリューションエクスプローラー]から開きたいcppファイルをダブルクリック
- 4 コードエディタにファイルの内容が表示される

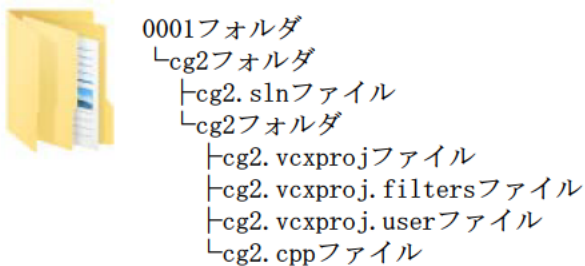


Visual C++のプロジェクトを使い回すときの注意点

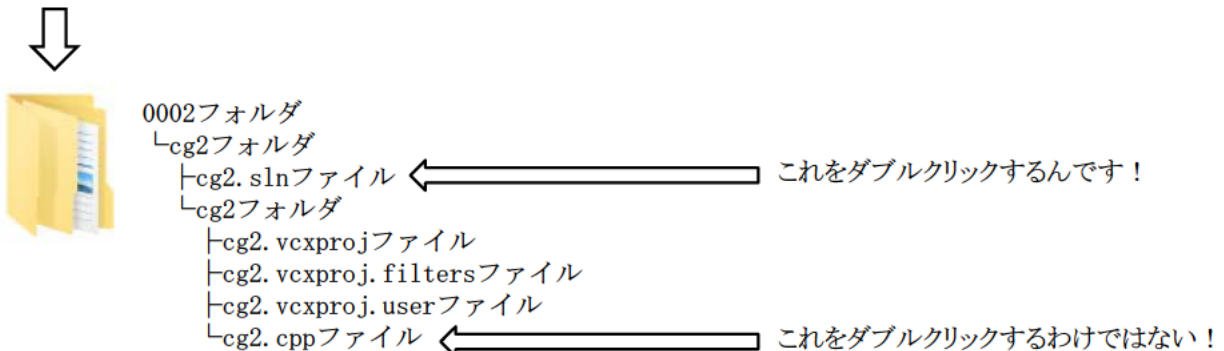
- ✖ フォルダごとコピーするのであって、.cppファイルを単体でコピーするわけではない
- ✖ フォルダ名を変更するのであって、.cppファイル名を変更するのではない
- ✖ サブフォルダ名を変更するわけではなく、全てのファイルの含まれているフォルダ名を変更する
- ✖ 前回編集したcppファイルを開くのであって、新しくcppファイルを作るわけではない
- ✖ 前回編集したcppファイルそのまま使うのであって、ファイル名を変更するわけではない
- ✖ Visual C++内の[ソリューションエクスプローラー]に表示されているcppファイルを開くのであって、エクスプローラにあるcppファイルを開くわけではない
 - ✖ ソリューションエクスプローラーが表示されていない場合は、メニューの[表示]-[ソリューションエクスプローラー]で開く



ファイル名を変えるわけではありません!!!
フォルダごとコピーしてフォルダ名を変えるのです!!!
エクスプローラのcppファイルをダブルクリックするわけではありません!!!
slnファイルをダブルクリックするのです!!!
cppファイルを新たに追加するわけではありません!!!
ソリューションエクスプローラーのcppファイルをダブルクリックするのです!!!



このフォルダ内にあるどのファイルも必要です
 このフォルダごとコピーして最上位のフォルダ名だけを変えるわけです



コンパイルエラー&リンクエラー

- 🔴 [出力]ウィンドウに[1 失敗]と出たらコンパイルかリンクが失敗
- 🔴 errorの内容を見て、どこにどんなバグがあるか確認する(この場合はmyInitとすべき所をmylnitと書いてしまった)

```
出力
出力元の表示(S): ビルド
1>----- ビルド開始: プロジェクト:Test, 構成:Release Win32 -----
1> Test.cpp
1>Test.cpp(39): error C3861: 'myInit': 識別子が見つかりませんでした
===== ビルド: 0 正常終了、1 失敗、0 更新不要、0 スキップ =====
```

```
void mylnit(char *programe){
{
    glutInitDisplayMode(GLUT_RGBA);
    glutInitWindowSize(500,400);
    glutInitWindowPosition(0,0);
    glutCreateWindow(programe);
    glClearColor(1.0,1.0,0.0,0.0);
}
```

- 🔴 [未解決の外部参照]関係のエラーや.libファイル関係のエラーはライブラリの設定がおかしい可能性がある
- 🔴 プロジェクトのプロパティに間違いがないかよく確認すること
- 🔴 ディレクトリをきちんと設定したかどうか、ディレクトリ名に誤字・脱字がないか確認すること
- 🔴 「¥」などの文字コードが異なっていないか確認すること

```
出力
出力元の表示(S): ビルド
1>Test.obj : error LNK2001: 外部シンボル "_imp__glutInitWithExit@12" は未解決です。
1>Test.obj : error LNK2001: 外部シンボル "_imp__glutInitWindowPosition@8" は未解決です。
1>Test.obj : error LNK2001: 外部シンボル "_imp__glutInitDisplayMode@4" は未解決です。
1>Test.obj : error LNK2001: 外部シンボル "_imp__glutKeyboardFunc@4" は未解決です。
1>Test.obj : error LNK2001: 外部シンボル "_imp__glutMainLoop@0" は未解決です。
1>Test.obj : error LNK2001: 外部シンボル "_imp__glutInitWindowSize@8" は未解決です。
1>Test.obj : error LNK2001: 外部シンボル "_imp__glutDisplayFunc@4" は未解決です。
1>D:\temp\GLintro\Release\Test.exe : fatal error LNK1120: 8 件の未解決の外部参照
===== すべてのリビルド: 0 正常終了、1 失敗、0 スキップ =====
```

```
出力
出力元の表示(S): ビルド
1>----- すべてのリビルド開始: プロジェクト:Test, 構成:Release Win32 -----
1> stdafx.cpp
1> Test.cpp
1>LINK : fatal error LNK1104: ファイル 'glut32.lib' を開くことができません。
===== すべてのリビルド: 0 正常終了、1 失敗、0 スキップ =====
```

- 🔴 基本的には、エラーメッセージを見てもどこにどんなバグがあるのか分からないことのほうが多い
- 🔴 カンを働かせつつシラミ潰しに一つ一つ詳細に確認するしかない
- 🔴 例えば、こんなエラーが出た場合

```
出力
出力元の表示(S): ビルド
1>----- すべてのリビルド開始: プロジェクト:Test, 構成:Release Win32 -----
1> stdafx.cpp
1> Test.cpp
1>C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\include\stdlib.h(376): error C2381: 'exit': 再定義; __declspec(noreturn) が異なります。
1> C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\include\GL\glut.h(146): 'exit' の宣言を確認してください。
1>Test.cpp(30): error C3861: 'exit': 識別子が見つかりませんでした
===== すべてのリビルド: 0 正常終了、1 失敗、0 スキップ =====
```

- 🔴 「stdlib hやglut hを修正すればいいのかな」などと考えるはいけない
- 🔴 既存のライブラリは修正しないこと
- 🔴 ちなみに、エラーの原因は「GL/glut hを先にincludeしてstdlib hを後でincludeしてた」なので「stdlib hを先にincludeしてGL/glut.hを後でincludeする」ことで解決する
- 🔴 エラーメッセージを見ても直接の解決法が書かれていることはほとんどないので、エラーメッセージを鵜呑みにせず、多面的に考えて解決法を探ること

- 🔴 [1 正常終了]と出たらコンパイル&リンクが成功
- 🔴 これで実行できる
- 🔴 実行できるからといって「実行時エラー」や「エラーメッセージの出ないバグ」がないとも限らない

```
出力
出力元の表示(S): ビルド
1>----- すべてのリビルド開始: プロジェクト:Test, 構成:Release Win32 -----
1> stdafx.cpp
1> Test.cpp
1> コード生成しています。
1> コード生成が終了しました。
1> Test.vcxproj -> D:\temp\GLintro\Release\Test.exe
===== すべてのリビルド: 1 正常終了、0 失敗、0 スキップ =====
```

うまくコンパイルできない！そんなときは

- 5 ビルドではなくリビルドを試してみよう
- 5 「pdbファイルに書き込めない」旨のメッセージが出た場合は、Visual Studioをいったん閉じて再び開くと直ることが多い
 - 5 ただ単に「pdbファイルに書き込めない」というメッセージが出るだけの場合、特に問題はない
 - 5 それでもうまくいかないのはVisual Studioではなく作成したプログラムに間違いがある可能性が高い
- 5 「idbファイルに書き込めない」旨のメッセージが出た場合は、同じソリューションファイルを二重に開いている可能性があります。使っていないVisual Studioは全て終了させてください。
- 5 「マルチバイト文字列」を使う設定にしてみよう
 - 5 [プロジェクト]-[○○○のプロパティ]をクリック
 - 5 [構成プロパティ]を選択
 - 5 [構成]で[すべての構成]を選ぶ
 - 5 [構成プロパティ]-[詳細]-[文字セット]で、[マルチバイト文字セットを使用する]に設定する
- 5 ライブラリを使用しているときにうまくビルドできない場合
 - 5 そのライブラリのhファイルや.libファイルへのパスが正しいかプロパティで確かめてください
 - 5 .libファイルの設定も正しいかプロパティで確かめてください
 - 5 #includeで.hファイルを正しくインクルードできているか確かめてください。
- 5 「LNK2026: モジュールはSAFESEHイメージには安全ではありません」というエラーが出る場合
 - 5 [プロジェクト]-[○○○のプロパティ]をクリック
 - 5 [構成プロパティ]を選択
 - 5 [リンカー]の[詳細設定]を開く
 - 5 [安全な例外ハンドラーを含むイメージ]で[いいえ]に設定する
- 5 「C2513: 指定された型は識別子の名前を指定せずに宣言されています」というエラーが出る場合
 - 5 "near"や"far"など、他のヘッダファイルで既に宣言されている変数・定数を利用していることが原因
 - 5 別の変数名にする

自宅でプログラミング

- 📌 自宅のWindowsパソコンでプログラミングしたい場合
 - 📌 Visual Studioを無料で手に入れる
 - 📌 Visual Studio Community 2019
 - 📌 <https://visualstudio.microsoft.com/ja/>
 - 📌 GLUTをインストールする(無料)
 - 📌 <http://www.transmissionzero.co.uk/software/freelut-devel/>
 - 📌 freelut 3.0.0 MSVC Packageを使う

GLUTのインストール

- 📌 ダウンロードして好きなフォルダ(例えばC:\Appsなど)に解凍
- 📌 freelut\binにあるfreelut.dllをC:\Windows\SysWOW64にコピー
- 📌 この例の場合、インクルードディレクトリはC:\Apps\freelut\includeになる
- 📌 この例の場合、ライブラリディレクトリはC:\Apps\freelut\libになる

仮想環境

- 📌 仮想環境は以下の2つ
 - ██
 - ██
- 📌 VMware Horizon Clientでログイン直後に「リモートコンピュータへの接続が終了しました」とすぐ終了するのはバージョンが古いから












要注意ポイント【コピペ編】

- ✖ 「このファイルのUnicode文字の一部は現在のコードページでは保存できません。データを保持するためにUnicodeで保存しなおしますか？」というエラーが出る場合
 - ✖ 通常は使われない特殊な文字が含まれている。PDFファイルなど、普通のテキストファイルとは異なるファイルからコピペしたときに発生する。
 - ✖ コピペをせず手動で入力する
 - ✖ [キャンセル]をクリックして、問題の箇所を探して修正する
- ✖ 「ユニバーサル文字名'¥u00A5'によって表示されている文字は、現在のコードページ(932)で表示できません」といったエラーが発生する場合
 - ✖ その行に特殊文字が存在するので、修正する
 - ✖ 行内のどの文字か分からない場合は1行まるごと書き直す
- ✖ 「定数値が切り捨てられました」というエラーが発生する場合
 - ✖ char型の1文字の定数はシングルクォーテーションマークで表現するが、'ab'のように2文字以上を指定すると発生するエラー
 - ✖ また、例えば'¥0'のようなときに¥マークが文字化けしていた場合、コンパイラが認識できない謎の文字「¥」1文字と文字「0」1文字の合計2文字として見なされる

要注意ポイント【Visual C++編】

- ✖ ビルドしたら「ソースファイルを開けません」というエラーが出る場合
 - ✖ ソリューションファイルしか存在せずcppファイルが存在しない, cppファイルが正しい位置にはなく別のフォルダに入っている, cppファイルのファイル名を変えた
 - ✖ ソリューションを作成したときと同じファイル構成, 同じファイル名じゃないときに発生するエラー
- ✖ メニューに[ビルド]の項目がない
 - ✖ ソリューションファイルを開かずにcppファイルを開いてしまった
- ✖ 修正したcppファイルの修正が反映されずに過去のcppファイルがコンパイルされているようだ
 - ✖ そのソリューションファイルに関連づけられていない別のcppファイルを修正している
 - ✖ ソリューションエクスプローラのcppファイルをダブルクリックして出てきたcppファイルを修正する

要注意ポイント【 編】

- ✖ 
 - ✖ 

 - ✖ 
 - ✖ 
 - ✖ 不要なファイルは削除する
 - ✖ "Debug"フォルダは不要
 - ✖ "Release"フォルダはexeファイル以外不要
 - ✖ pdb/ipdb/ilk/ipch/idb/pch/obj/iobj/log/lastbuildstate/tlog/suo/dbファイル等は不要
 - ✖ sdfファイルや.vsフォルダも不要. これらは必要なファイルだが, Visual Studioを起動すると自動的に作られるので, 消しても問題ない
 - ✖ 自分の作ったプログラムは残しておくこと. 実行するために重要なファイルである sln/vcxproj*/h/cppファイルは残しておくこと
 - ✖ ファイルは完全に削除する. ゴミ箱にファイルが残っている場合はゴミ箱を空にする
 - ✖ 


- ✖ 
 - ✖ 
 - ✖ 

要注意ポイント【基本編】

- ✖ ソースコードに出てくる「//」は「それ以降の行末までをコメントアウトする」という意味の、C++でのコメントを表している
- ✖ for文の中で使っている変数がおかしいバグに注意すること
 - ✖ for(i=...
for(i=...
など
- ✖ if(a==3)と書くつもりでif(a=3)と書いてしまわないように注意
- ✖ glutInitってちゃんと書いたはずなのにコンパイルできない場合
 - ✖ glutnitではなくglutInitが正しい。アルファベットの小文字のエルと大文字のアイは間違えないように注意する。
- ✖ 配列に初期値を代入するところで{}の中カッコを使うはずが()の丸カッコを使っていないか注意
- ✖ x / Wのような計算が想定した計算結果にならない場合
 - ✖ xがint型で0以上W未満の値をとり、Wがint型の定数だとする
 - ✖ x / Wのように計算して0以上1未満の実数値を計算したかたとする
 - ✖ int型同士の計算なので小数点以下が無視される
 - ✖ この場合、常に0という計算結果になる
 - ✖ double型に変換してから計算させる必要がある
 - ✖ この場合、(double)x / (double)Wのように実装する

#includeの赤線

- 🔍 プログラムのおかしい箇所にVisual Studioが自動的に赤い波線を書いてくれるためバグを見つけやすい

```
void myKeyboard(unsigned char key)
{
    ifu (key == 0x1B) exit(0);
}
```

- 🔍 赤い波線が直接の原因とは限らない
 - 🔍 上の例では「exit」には問題がなく「ifu」に問題がある(正しくは「if」)が、「ifu」が間違っているため、それ以降のソースコードをうまく解析することができなくて、「exit」までも余計に「間違っている」と判断されてしまっている(「ifu」が関数名として判断されてしまっている)
- 🔍 #include <gl/glut.h>に赤線がある場合
 - 🔍 プロパティの設定が間違っている
 - 🔍 資料の通りに入力したか? 「¥」マークをコピーではなくちゃんと手動で入力したか?
 - 🔍 フォルダごとコピーすること
 - 🔍 ファイルだけコピーしたりしていないか? ファイルをファイルエクスプローラ上でダブルクリックしていないか? ソリューションファイルをダブルクリックして開いたか? ソリューションエクスプローラ上のファイルを開いているか?
 - 🔍 ソリューション構成とソリューションプラットフォームはプロパティ設定時と同じ設定になっていること
 - 🔍 [ソリューション構成]を[Debug]にして、[ソリューションプラットフォーム]を[x86]にすれば直ったりしないか?

エラー

- 🔍 LNK1201 Visual Studioを全て閉じてもう一度開き直す
- 🔍 C1033 Visual Studioを全て閉じてもう一度開き直す
- 🔍 C1041 Visual Studioを全て閉じてもう一度開き直す
- 🔍 LNK2001 円マークは正しいか？ディレクトリを手動で再度書き直す
- 🔍 LNK1120 円マークは正しいか？ディレクトリを手動で再度書き直す
- 🔍 LNK1104 円マークは正しいか？ディレクトリを手動で再度書き直す
[ソリューションプラットフォーム]を[x64]ではなく[x86]にする
リビルドする
- 🔍 C1083 円マークは正しいか？ディレクトリを手動で再度書き直す
[ソリューションプラットフォーム]を[x64]ではなく[x86]にする
- 🔍 LNK2005 cppファイルを追加するのではなく、以前に作ったcppファイルの内容を上書きする
- 🔍 C4566 文字化けを直す
- 🔍 C4309 ‘¥0’などで¥が文字化けしているので¥を手動で書き直す
- 🔍 C2381 stdlib hを先にincludeしてGL/glut.hを後でincludeする
- 🔍 C3861
 - 🔍 文字の間違い(大文字のアイ・小文字のエル・数字のイチ, など)
 - 🔍 stdio hなどincludeすべきファイルがincludeされていない
- 🔍 C2513 その変数・定数は既に定義されているので別の変数名にする
- 🔍 C2872 インクルードする順番を変えてみる
- 🔍 LNK1235 「マネージ拡張」を「いいえ」にする
- 🔍 LNK2026
 - 🔍 [構成プロパティ]-[リンカ]-[詳細設定]-[安全な例外ハンドラーを含むイメージ]で「いいえ」を選ぶ
- 🔍 C4996
 - 🔍 [構成プロパティ]-[C/C++]-[全般]-[SDLチェック]で「いいえ」を選ぶ
 - 🔍 ソースコードのどこか(上のほう, など)に以下の行を追加する
#pragma warning(disable:4996)

確認メモ

- 🔍 初回の授業で、Visual Studioを起動し、赤い四角形を表示するまでの作業は、資料の通りに行動すれば誰でも同じようにうまくいく
 - 🔍 実際、周りを見てみれば、ほとんどの受講学生が出来ている
 - 🔍 つまり、うまくいかないのは受講学生が資料の通りの行動をとっていないことが原因
- 🔍 インクルードディレクトリ・ライブラリディレクトリを正しく入力できていない
 - 🔍 PDFファイルのディレクトリをそのままコピーすると、¥マークが文字化けする
 - 🔍 [OK]を押したつもりが[キャンセル]を押した場合、受講学生本人は正しく設定したつもりが、実際は設定されていないということが起こる
 - 🔍 ソースコードを見て、#includeに赤い波線があるかどうか確認しよう
 - 🔍 エラーメッセージに.libファイル名やOpenGLの関数名があるかどうか確認しよう
 - 🔍 ディレクトリ名は手動で全て入力しなおそう
- 🔍 「ファイルを追加するわけではない」「ファイル名やサブフォルダ名は変えない」「フォルダごとコピーする」ということを無視して、ファイル単体をダブルクリックしたり、ファイルを追加したり、ファイル名やサブフォルダ名を変更することで問題が発生することも多い
 - 🔍 ソリューションエクスプローラのcppファイルをダブルクリックしてみよう
 - 🔍 開かれたcppファイル以外のcppファイルを閉じて、ソリューションエクスプローラから削除しよう
 - 🔍 slnファイルを開いたときにソリューションエクスプローラに何も表示されない場合(読み込み失敗)、その原因の可能性の一つが「ファイル名やサブフォルダの名前を変えた」
 - 🔍 「新しくプログラムを作ったのに新しいプログラムが実行されずに前回作ったプログラムが実行される」の原因も「ファイルを追加した」という可能性が高い
- 🔍 「Visual Studioを全て閉じて」の際、ゾンビプロセスとして残ったVisual Studio関連の各プロセスも殺すこと
- 🔍 「ビルド」ではなく「リビルド」をする
- 🔍 どうしても解決できない場合は再ログインしてから、初回の授業の資料の通りに最初からやり直したほうが確実