高知大学教育学部の情報数学のテキスト 文責: 高知大学名誉教授 中村 治

おまけ: 数独のヒントを表示してくれるプログラム

加筆修正:visual studio 2019 でコンパイルする方法を追加しました。

これは平成23年に、一年生用の「学問基礎論」という講義で、プログラミングの例として作 り、学生さんに実習してもらったものに加筆したものです。私の講義の中では、多分、唯一、好評 だったものです。これを拡充すれば、数独をコンピュータに解かせたり(単に解くだけならバック トラッキングで簡単ですが、人間の各種解法をシミュレーションしたり、そのためのヒントを表示 したりするとプログラムは随分長くなります)、数独の問題を作るプログラムに育て上げることが 出来ます。但し、私の作ったプログラムは二万行ぐらいあります。「問題作成」のボタンを押せば



盤面に表示されている図のような問題を作ってくれます。これは乱数を使って、コンピュータにラ ンダムに作らせた問題です。図の数字の配置は対称ではありませんが数字の配置が対称なものを普 通「数独」と言っているみたいです。このプログラムは、「問題作成」のボタンを作る前にほとん どを作ったプログラムで、数独の解法のプログラミングのデバッグのために使ったサンプルにして いる「数独の問題」に著作権があるので、そのままでは公表出来ませんし、数独を解く楽しみが無 くなります。職業プログラマでない私のようなものがこのようなプログラムを作成するためには、 かなりの量のサンプルプログラムが必要です。数独の場合のように人間のエクスパートが使ってい る解法が大量に公表されているパズルでもエクスパートが使っている解法をプログラミングする場 合、論理的に考えて、数学的に間違いがないと思っても、実際は見落としがいっぱいあって、その 解法が必要な問題にあったって、延々とバグを取り除く必要があります。問題作成はある意味独立 に出来ますが、人間のエクスパートが使っている解法をプログラミングしておかないと、問題作成 に必要な解が無いか、一つか二つ以上かを効率よく判定するプログラムを作れませんし、また作成 した問題の難易度が自分で解いてみないと分からなくなります。個人が趣味で作るプログラムとし ては必然的にかなり大型のプログラムになります。私は、自分で解くのがめんどくさいので、「ひ とりにしてくれ」、「ぬりかべ」、「ましゅ」、「カックロ」、「フィルオミノ」、「四角に切れ」「スリ ザーリンク」、「美術館」、「へやわけ」、「のりのり」などをコンピュータに解かせたり、解法の技 法を探したり、問題作成はどのようにすればできるか探索して楽しんでいます。

普通の人は単に目についたパズルを解いてみます。そして、一部の人はパズルの収集を始めま す。解くことよりも収集が目的となります。数独など一部のパズルは発表されているパズルの本や 雑誌が大量で収集は個人では大変です。この病気が進むと、最後は、パズルの問題を自分で作るよ うになります。数独をヨーロッパで広めた人は数独の問題を作るプログラムを作るのに数年を要し たと何かで読みましたが、現在は数独が欧米で数学的に調べ尽くされているのでその知識を使い、 問題の質を問題にしなければ、数独の問題をコンピュータに作らせるのは簡単です(数学の入試問 題の作り方と同じことをすれば良いです。つまり、解答の図面をランダムに作り、ランダムに数字 を消していって、解が唯一決まる極小のものを問題とすれば良いです。「問題作成(点対称)」のボ タンを押せば



のような普通の数独の問題を、「問題作成(X軸対称)」のボタンを押せば



のような上下対称な問題を、「問題作成(Y軸対称)」のボタンを押せば



のような左右対称な問題を作ってくれます。更に、「mask 作成」で数字を表示するセルを指示し、 問題が作れそうな数字達を適当に配置し(コンピュータにランダムに配置させることも勿論可能で す)、「問題作成 (mask)」のボタンをクリックすれば、運が良ければ(上に述べた極小の配置を求 めることは「山登り法」を使えば常に可能ですが、これは「山登り法」と遺伝的アルゴリズムとい う不確実なアルゴリズムの組み合わせを使っているので、残念ながら確実に問題が作れる方法では ありません。インターネットには、良いアルゴリズムがあると書いている人もいますが、私には残 念ながら、その方法が分かりません。)



のような問題を作ってくれます。この問題は超難問です。この最後の方法は「数独」の問題の空間 の位相構造が極めて特殊だから出来ることで、その他の鉛筆パズルに応用することは残念ながらで きません。

作る問題の難易度は乱数で作っているので、勿論出鱈目です。

そして、このような問題を大量に作り、良いものを選択すれば良いです。人間が介入し、特定の 技法を実現するようにすればよい問題が作れると思います。但し、私はそのためのアルゴリズムを 知りません。良い問題とは何かの基準が分かりません。ある人にとって良い問題でも、他の人に とっては難しすぎて、あるいは易しすぎて良くない問題であり得ます。そして、その経験を積めば、 何が本質的か理解できるようになり、人間の介入も外すことが出来るはずですが、そのようなこと まで考慮するとやるべきことはいくらでもあります。所謂、人工知能の問題になります。)。同じ アイデアで問題が作れるパズルもありますが、一般的にはニコリのパズルの解く価値のある面白い 問題をコンピュータで人間の介入なしに作るのは結構難しいです。勿論、数百種類?ある全部のパ ズルを確かめたわけではありませんが。兎も角、私が、問題作成のプログラムを作ってみたのは、 「ひとりにしてくれ」



と「へやわけ」



と「のりのり」



と「スリザーリンク」



と「ぬりかべ」



と「フィルオミノ」(但し、単純なアルゴリズムを使っているので、数字の少ない綺麗な問題を作ってくれない!この pdf の後半に問題点を解説しています。)

											1 3% 25 million 100 million
	2		4	3	4		1	3		5 横	10
	5									数值入力	SpecialD
5		5	З			9		1		Reset	SpecialD On
	5			З			8	8		First Step Simple	SpecialE
4		5				8			4	Simple Only	SpecialF
			1	4	9	5	1		8	SpecialA	SpecialF Or
3	2	4	1		4			8		SpecialA Only	
-	-	-	-	3	2		+	4	4	SpecialB SpecialB Oply	
				-	_		3	4	4	SpecialC	問題作成
	3	1	-	3			3		-	SpecialC Only	Backtrackin (More Than Two)
				0			<u> </u>			label 1	

と「シャカシャカ」







と「四角に切れ」(この pdf の後半に問題の作り方を解説しています。)



です。表示している問題はソフトが作った問題の一例です。年金生活ですから暇ですが、他の沢山 ある趣味の「お勉強」にも多くの時間が割きたいから、暇を見て、その数を増やしています。

さて、前置きはこれくらいにして、数独のヒントを表示するプログラムを VC++で作ってみま しょう。今までと同様に、Microsoft Visual Studio 2010 を立ち上げ、スタートページを表示する。 スタートページのタグがなければ、表示のメニューから「スタートページ」をクリックする。

ファイル(E) 編集(E) 表示(Y) Qt5 デバッグ(D) チーム(M)	テーダ(A) ツール(I) テスト(S) ウィンドウ(W) ヘルプ(H)
🚰 • 🕮 • 😂 🖉 🗶 45 (55 19 • 19 • 19 • 19	
	<u>a</u>
スタートページ × ソリューション エクスプローラー クラス ビュー	- · · ·
4	
Microsoft*	
Visual Studio [®] 2010 Profess	ional
DB	
4	作業の開始 ガイダンスとリソース 最新ニュース
Team Foundation Server への接続	
「「「新しいプロジェクト	$z \rightarrow z \leftarrow Windows Web (777)^{-1} Office SharePoint \tau = 9$
「「「」 プロジェクトを聞く	Vicual Studio 2010 096889
	くいちは State
目いにはったゴロントクト	Visual Studio 2010 の概要
TELLIE SEJUSISI	····································
3 sudoku	Wisual C++ の時代機能 Visual C++ の時代機能 Visual C++ の時代機能
HITORI	
SUMOFKETA	
 ✓ フロジェクト読み込み後にページを閉じる ✓ スタートアップ時にページを表示する 	Visual Studio を使用したアプリケーションの作成
出力	+ 9 ×
出力元の表示(5):	- 0 0 2 3 1
港供完了	

新しいプロジェクトをクリックし、 Window フォーム・アプリケーションをクリックし、名前に sudoku を打ち込み、

		新しいプロジェクト		? ×
最近使用したテンプレー	۲. N	NET Framework 4 Y 並べ替え基準: 既定	~ III III	インストールされたテンプレートの検索
インストールされたテンプレート Qt5 Projects		Win32 コンソール アプリケーション	Visual C++	種類: Visual C++ Windows ユーザー インターフェイスを含むアプリ
✓ Visual C++ ATL CLD	ú	MFC アプリケーション	Visual C++	ケーションを作成するためのプロジェクトです。
全般 MEC		Win32 プロジェクト	Visual C++	
テスト Win32		空のプロジェクト	Visual C++	
▷ 他の言語 ▷ その他のプロジェクト	の種類	ATL プロジェクト	Visual C++	
 >> データペース >> テスト プロジェクト 	1	MEC DLL	Visual C++	
オンライン テンプレート		Windows フォーム アプリケーション	Visual C++	
	l.	CLR コンソール アプリケーション	Visual C++	
	1	☆ 空の CLR プロジェクト	Visual C++	
名前(≥):	sudoku			
場所(L): C:¥texsrc¥情韩		¥集中講義¥sudoku¥	~	参照(B)
ソリューション名(<u>M</u>):	sudoku			> ソリューションのディレクトリを作成(D) > ソース管理に追加(U)
				OK キャンセル

OK ボタンをクリックする。

sudoku - Microsoft Visual Studio Academic	- 1	8 ×
ファイル(E) 編集(E) 表示(V) Qt5 プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) ツール(I) テスト(S) ウィンドウ(VV) ヘルプ(H)		
i 👔 • 😂 • 😂 🛃 🥔 🖇 🖧 🕫 - 😌 • 😂 - 🎉 🕨 Debug 🔹 Win32 🔹 🔹 🙆 abs 🔹 🔹 🔩 🔂 💥 🐼 🗮 🗮 🗖 •		
· \$P\$ \$P\$ \$P\$ \$P\$ \$		
◎ Form1.h [デザイン] × ソリューション エクスプローラー クラス どユー -	プロパティ	+ # ×
	Form1 System.Window	/s.Form +
	Form1 System.Window Televise TimeNode Nocont TimeNode Nocont TimeNode Nocont TimeNode Nocont AutoScrolle (5, 0) MinimumSis (5, 0) MinimumSis (5, 0) MinimumSis (5, 0) State 300, 20 StateThostare Mindow MindowState Norma Consort Color (1) Background [1] Carsor Default (1) Carsor	nly o o vsDefat d mtrol EU)
	Font MS UI	Gothic,
	FormBorder: Sizable RightToLeft No RightToLeft False Text Form1 UseWaitCurs False	v

Microsoft Visual Studio 2010 では、Window フォーム・アプリケーションを作るのはこのよう に簡単でしたが、Microsoft Visual Studio 2013 では、Window フォーム・アプリケーションの選 択肢がなくなり、「猫でもわかる Windows プログラミング」にあるような昔ながらのコードによる Window フォーム・アプリケーションは作れますが、Microsoft Visual Studio 2010 で普通にやっ ていた手軽さに慣れてしまうと Microsoft Visual Studio 2013の選択肢にあるような複雑なソフト ではなく、単にこのような単純なソフトを作りたい素人は途方にくれます。幸い、VC++ 2013 で Windows フォームアプリケーションのプログラムを作る方法をインターネットで見つけたので、 かなり複雑ですが、やってみます。

「新しいプロジェクト」で、「Visual C++」の「CTR」を選択し、「空の CLR プロジェクト」を 選択し、「名前」を指定し、

			新しいプロジェクト		? ×
▶ 最近使用したファイル		.NET F	ramework 4.5 👻 並べ替え基準: 既定	• # E	インストール済み テンプレート の検索 (Ctrl: 👂 -
▲ インストール済み		则节	クラス ライブラリ	Visual C++	種類: Visual C++
▲ テンプレート ▷ Visual Basic ▷ Visual C#		2431 C/	CLR コンソール アプリケーション	Visual C++	ローカル アプリケーションを作成するための空のプ ロジェクトです。
✓ Visual C++ Windows	<u> </u>	2,1	空の CLR プロジェクト	Visual C++	
ATL CLR 全般 MFC テスト Win32 Visual F# SQL Server TypeScript JavaScript Python ト このがつていた。	ちんの領域賞 マ		オンラインでテンプレートを検索するには、ここをかいめしまで	5.	
* 12312					
名前(<u>N</u>):	WinFormApp				
場所(<u>L</u>):	C:¥vcwinprog¥[Dialog¥		•	参照(<u>B</u>)
ソリューション名(<u>M</u>):	WinFormApp				 ✓ ソリューションのディレクトリを作成(<u>D</u>) □ ソース管理に追加(<u>U</u>)
					OK キャンセル

「OK」ボタンをクリックする。

♥ WinFormApp - Microsoft Visual Studio ブイル(E) 編集(E) 表示(Y) プロジェジト(E) ビルド(E) デバッグ(D) チーム(M) ツール(I) デスト(S) 分析(M) ウインドウ(M) ヘルプ(H) ※ O ・	】 ▼5 クイック起動 (Ctrl+	Q) P - 西 X 中村治 - 🏢
9-1 1020- 3-	yun=>ay x0x7u=== ① ○ ・	- 東 × 「御 / 」 没様素(と(t++:) App' (1 プロジンクト)
	プロパティ	- ą ×
	WinFormApp プロジェクトの	Dプロパティ -
出力		
出力元の表示(5): 🔹 🔹 🛬 🖆 🖆	(名前)	WinFormApp
	プロジェクト ファイル	C:¥vcwinprog¥Dialog¥WinForr
	プロジェクトの依存関係	
	ルート石前空間 (名前)	wineormapp
	フロジェクト名を指定します。	
1万-一覧 出力	ツールボックス <mark>プロパティ</mark>	
<u>事借完了</u>		

右側の「ソリューションエクスプローラー」で、WinFormApp(プロジェクト名)を右クリック し、「追加」->「新しい項目」をクリックする。左欄で「Visual C++」->「UI」を選択し、右 欄で、「Windows フォーム」を選択して、

		新しい項目の追	自加 - WinFormApp	? ×
▲ インストール済み		並べ替え基準: 既定	• # E	インストール済み テンプレート の検索 (Ctrl・ 🔎 🗸
▲ Visual C++ Windows スト	P	E Windows フォーム	Visual C++	種類: Visual C++ 他の Windows コントロールを含む CLR フォー
UI HLSL		CLR ב-ザー כארם-א	Visual C++	ムを作成します。
コード データ リソース Web ユーティリティ プロパティ シート Test		MFC リボン定義 XML ファイル	Visual C++	
▶ オンライン		オンラインでテンプレートを接	素するには、ここをクリックします <u>。</u>	
名前(<u>N</u>): 場所(<u>L</u>):	MyForm1.h C:¥vcwinprog¥[Dialog¥WinFormApp¥WinFormApp	•]	参照(<u>B</u>) 追加(<u>A</u>) キャンセル

「追加」をクリックする。名前は「MyForm.h」となっていて、従来の Form1.h に相当する。

♥ WinFormApp - Microsoft Visual Studio ワ アイル(E) 編集(E) 表示(Y) プロ5±2か(E) ビルド(E) デバッグ(E) チーム(M) ツール(II デスト(S) 分析(M) ウィンドク(M) ヘルプ(H) ◎ • ○ ○ ◎ 〒 ・	▼5 クイック起動 (Ctrl+Q)	♪ ■ ♂ × 中村治 ~
Y MyForm □ Image: Second sec	ソリューション エクスプローラー ・ ・ ・ ・ ・ ・ ・	 + + × # (ст(+:) p (1 ЛЭзгор) (1 ЛЭзгор)
	プロパティ WinFormApp プロジェクトのプ 2011 日 クロジェクトのプ	- ↓ × □/(ティ -
出力 出力元の表示(<u>S</u>): ・ 全 雪 雪 竺 ね	 その他 (名前) プロジェクト ファイル プロジェクトの依存関係 ルート名前空間 	WinFormApp C:¥vcwinprog¥Dialog¥WinForr WinFormApp
15 5 出力 全長元了	(名前) プロジェクト名を指定します。 ツールボックス プロパティ	

「MyForm.h」ができ、フォームデザイナに、Windows フォームが表示される。

次に、エントリーポイントを作成します。エントリーポイントとは、プログラムを起動したとき、 最初に実行するプログラムであり、具体的には main() 関数のことである。右側の「ソリューション エクスプローラー」で、「ソースファイル」を展開し、「MyForm.cpp」を表示する。「MyForm.cpp」 を右クリックし、「名前の変更」を選択して、「WinFormApp.cpp」(WinFormApp.: プロジェク ト名)に書き換える。

WinFormApp - Microsoft Visual Studio アイル(E) 構築(E) 表示(Y) プロジェかト(E) ビルド(B) デパッグ(E) チーム(M) ツール(E) デスト(S) 分析(M) ウィンドウ(M) ヘルプ(H) ◎ ● ● (昭・ வ 日 ●) ワ・ ペ・ ▶ ローカル Windows デパッガー・ ○ • [Debug •] 勇 。※ 本 ト キ コ 下 井 山 肉 工 話 ⊕ 井 キ ば ぼ 。	▼5 クイック起動 (Ctrl+Q)	P - P × 中村治・日
MyForm C X	ソリューション エクスプローラー ・ ・ ・ ・ ・ ・ ・	- 9 x - 0- 0F) - 052 K2-
	プロパティ WinFormApp.cpp ファイルのプロパティ	+ 4 × ,
出力 出力元の表示(S): - ・ を き = 荃 ね ・ + × 5	■2: j ≠ 1 日 その他 (名制) WinFormAj コンテンツ False フッイルの理想 C/C++ コー プロジェクトに含まれる True (名崩)	pp.cpp
177	ファイル オブジェクトに名前を付けます。 ツールボックス プロパティ	

「ソリューションエクスプローラー」で WinFormApp (プロジェクト名)を選択した状態で、メ ニューから、「プロジェクト」ー>「プロパティ」をクリックする。

				-	yui-say i9x70-5- C C G G G To + ₹ f	· in -	- 4 ×
해가 해외 MyForm		WinFormApp		? ×	ソリューション エクスプローラー の マリューション 'WinForm MinFormApp	検索 (Ctrl+:) App' (1 プロジェクト)	ρ.
	(約次(C): IVA 参照 参照 ▶ 構成プロパティ ● 構成プロパティ	対象のフレームワーク: JNETFran 参照(れ): 名類 - OSystem.Data - OSystem.Data - OSystem.Drawing - OSystem.Unidows.F., - OSystem.Xml	ework,Version=v4.5 ・アセンブリ参加プロパティ ・思いアフロパティ ローカルのガビー・・メ・メ・メ・ドアのgro ローカルのガビー・False ローカルのガビー・False ローカルのガビー・False ローカルのガビー・True 参照アロパティ ID アセンブリを、CitProgram File アセンブリを、System, Version カルドディ ・アレーンプリを、 アセンブリを、 アセンブリを通びロパティ	s (x80)¥Refe =4.0.0.0, Ct		チーム エクスプローラー クラス プロパティ	5- 952 ピュ− - 마×
出力 出力元の表示(S):		新しい参照の追加(N)	第四の削除(E) OK キャンセル	↓ × 適用(A)	日 その他 (名前) プロジェクト ファイル プロジェクトの依存関係 ルート名前空間	WinFormApp C:¥vcwinprog¥Dialog WinFormApp	¥WinForr
17 11 出力					(名前) プロジェクト名を指定します。 ツールボックス プロパティ		

「WinFormApp(プロジェクト名)プロパティページ」が開くので、左の欄で、「構成プロパティ」->「リンカー」->「システム」と選択し、右の欄で「サブシステム」を選択し、右端に現われる「V」をクリックし、現われたリストから「Windows(/SUBSYSTEM:WINDOW)」を選択する。

	WinFormApp	プロパティ ページ	? ×
構成(C): アクティブ(Debug)	✓ プラットフォーム(P):	アクティブ(Win32)	▼ 構成マネージャー(0)
全般 デバッグ VC++ ディレクトリ ▷ C/C++ ▲ リンカー 全般 入力 マニフェスト ファイル デバッグ システム 最適化 埋め込み IDL Windows ソケニー	サブシステム 最低限必要なバージョン ヒープ サイズの設定 ヒープ コミット サイズの設定 スタックのサイズの設定 スタックのコミット サイズ 大きいサイズのアドレス ターミナル サーバー CD からスワップ実行 ドライバー	Windows (/SUBSY いいえ いいえ 設定なし	STEM:WINDOWS)
	サブシステム /SUBSYSTEM オブションは、オペI の選択によって、リンカーが選択する	ノーテイング システムに .exe ファイルム エントリ ポイント シンボル (またはエン OK	の実行方法を指定します。サブシステム トリ ポイント関数)が影響を受けます。 キャンセル 適用(A)

右下の「適用」をクリックする。

			WinFormApp 7	プロパティ ページ	? 🗙
構成(C):	アクティブ(Debug)		✓ プラットフォーム(P):	アクティブ(Win32)	▼ 構成マネージャー(0)
4 7 ∨ 0 C	一般 バッグ C++ディレクトリ /C++ シカー 全般 入力 マニフェストファイル デバッグ システム 毎週化 埋め込み IDL	^	サブシステム 最低限必要なパージョン とーブ サイズの設定 とーブ コミット サイズの設定 スタックのコミット サイズ スシックのコミット サイズ 大きいサイズのアドレス ターミナル サーバー CD からスフップ実行 ネットワークからスフップ実行 ドライバー	Windows (/SUBS) いいえ いいえ 設定なし	YSTEM:WINDOWS)
< N 7	Windows メタデータ 詳細設定 すべてのオプション コマンド ライン -フェフト ツー11. >	~	サブシステム /SUBSYSTEM オブションは、オペレ の選択によって、リンカーが選択するユ	ーティング システムに .exe ファイル こントリ ポイント シンポル (またはエン OK	の実行方法を指定します。サプシステム トリポイント関数) が影響を受けます。 キャンセル 適用(A)

つぎに、「構成プロパティ」->「リンカー」->「詳細設定」を選択し、右欄の「エントリポイント」に、「main」と入力して、

構成(C): アカティブ(Debug)
システム 対象コンピューター MachineX86 (/MACHINE:X86) 最適化 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・

「OK」をクリックする。

♥ WinFormApp - Microsoft Visual Studio ワイル(E) 編集(E) 表示(Y) プロジェガト(E) どルド(E) デパッグ(D) チーム(M) ツール(I) アスト(E) 分析(M) ウインドク(W) ヘルプ(H) ◎ • ○ 後・② 論 論 例 ワ・♡・ ▶ ローカル Windows デパップ・・○ ・ Debug ・ 声・□ 下 非 山 神 工 器 音 非 ≑ (ぷ ぷ)。	】 ▼5 りイック起動 (Ctrl+Q) P - 日 × 中村治 - 🏢
	ソリューション エクスプローラー	- ₽ ×
	© © ∰ `o - ≠ @	<u>ه ۲ –</u>
MyForm D 🖾	ソリューション エクスプローラー のれ	除索 (Ctrl+:) ・
	JULーション 'WinFormA	pp' (1 プロジェクト)
	WinFormApp	
	++ WinFormApp	.cpp
	▲ 🚍 ヘッダー ファイル	
p	ジョー リソース ファイル	
	▶ 👼 外部依存関係	
lb	ソリューション エクスプローラー 🗧	ーム エクスプローラー クラス ビュー
	プロパティ	- ¢ ×
	WinFormApp プロジェクトの	プロパティ ・
	11 gu 🖌	
	日 その他	
	(名前) プロジェクト ファイル	WinFormApp C:XvcwinprogXDialogXWinForr
	プロジェクトの依存関係	C.+veminprog+blulog+vini on
	ルート名前空間	WinFormApp
	(名前)	
	プロジェクト名を指定します。	
1万一一覧 出力	ツールボックス プロパティ	

「ソリューションエクスプローラー」で、「WinFormApp.cpp」(WinFormApp.:プロジェクト名) をダブルクリックする。

WinFormApp - Microsoft Visual Studio アイル(E) 編集(E) 表示(y) プロジェジト(E) デバッグ(D) チーム(M) ツール(I) デスト(S) 分析(M) タインドジ(M) ヘルプ(H) ○・○ 後・⑤ 単 単 ツ ・ ⑦・ ○ ト □ーカル Windows デバッパー・ ○・ ○ Debog ・ 輝 ・ □ 指 軍 強 東 領 領 領。	」 ▼5 ケイック起動 (Ctrl+Q) P = 日 × 中村治 - 日
Y WinFormApp.cpp * WinFormApp.cpp * (7D-/CVL Z3-7) * * Hinclude "WyForm-h" *	ソリューション エクスプローテー ・ Ⅱ × ● ▲ ● ・ # ● ● ● ● ソリューション エクスプローテー の装着 (Cri+1) P ・ □ ソリューション WinFormApp (1 プロジェクト) ・ ■ ▲ ③ ソース ファイル ・ ● ● ● ● ● ● ● ○ ● ○ ● ○ </td
100%。4 出力 出力元の表示(S):	レーション レンジョー テーション レーラー ウラス ビュー プロ(ティ ・ 非 × 「 「 「 」 「 」 「 」 「 」 「 」 「 」 「 」 「 」 「
25	ッ1xinyX <mark>プロ(ティ</mark> 1行 1列 1文字 棒入

ここに、

#include "MyForm.h"

using namespace WinFormApp;

```
[STAThreadAttribute]
int main()
{
```

Application::Run(gcnew MyForm());
return 0;

}

となるようにプログラムを打ち込む。







上手くいきました。

パソコンが急に壊れ、新しく購入したパソコンに Microsoft visual studio 2012 professional をインストールしてみましたが、うまく

いきませんでした。Microsoft visual studio 2019 community を デフォルトのままインストールすると「猫でもわかる Windows プ ログラミング」で解説されているような昔の「使って天国、作って 地獄」の時代のプログラミングしかできません。しかし、Microsoft visual studio 2019 community でも、以下のプログラムを作成で きる方法をインターネットで見つけました。こんな抜道を見つけ出 す人たちはどんな勉強をしてきたんでしょうか?教育に携わってき た人間として興味があります。

すべてを書き直すのは大変ですから、最低限の修正で済ませること にします。

まず Visual Studio の公式サイトから、Community エディションのインストーラーをダウンロードします。



実行をクリックして、インストールを始めます。途中の



の画面のページになったとき、デフォルトではなく、デスクトップとモバイル(5)の 「.NET デスクトップ開発」と「C++によるデスクトップ開発」と右側のオプショ ンの「v142 ビルドツール用 C++/CU サポート」にチェックを入れることを忘 れないでください。後はデフォルトでいいです。

まず、Microsoft Visual Studio 2019 を立ち上げる。

是近開	いた項目(民)		開始す	3
	<mark>،</mark> م		*	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
	winTest.sln	2020/07/11 0:25		
	C:¥C++src¥winTest		Ċ	プロジェクトやソリューションを開く(<u>P</u>)
2	winTest2.sln	2020/07/11 0:10		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
	C:¥C++src¥winTest2			
昨日			<i>d</i> -	ローカル フォルダーを開く(F)
4	whist.sln	2020/07/10 1:49		任意のフォルダー内のコードに移動して編集します
	C:¥texsrc¥情報数字¥集中調義3¥cardGame¥c#¥whist			
学過				
2	whist.sin C.Wayercy注起数学X信中注意2XcardGameXc#Xysbirttert	2020/07/09 10:53	Ð	新しいフロジェクトの作成(<u>N</u>)
	C.+texsic+ig to go T in gest condition text + whistlest			開始するには、コード スキャフォールティンク付きのプロジェ クト テンプレートを選択します
2	whist.sln	2020/07/09 7:29		
	G:#texsrc半情報数字半集中講義3¥cardGame¥c#¥whist			コードなしで続行(W) →
	ShogiTeX.sln	2020/07/05 12:09		

私は既に使っていて、左側にその履歴が残っていますが、それは気にしなくていいです。右の一 番下の「新しいプロジェクトの作成(N)」をクリックします。



この画面で、右側の選択肢の下の方にある「CLR 空のプロジェクト(.NET Framework)」を選択し、

最近使用したプロジェクト テンプレ	ノート(<u>R</u>)	すべての言語(L) ・ すべてのブラットフォーム(P) ・ すべてのプロジェクトの種類
ຮັງ CLR 2007ົບວັນງາໄ (NET Framework) ຮັງ 2007ົບວັນງາໄ	C++	WUnit テスト プロジェクト (.NET Core) Windows、Linux および MacOS 上の.NET Core で実行できる NUnit テストを含むプロジェクト。 デスクトップ F# Linux macOS アスト Web Windows ダイナミック リンク ライブラリ (DLL) 実行中の複数の Windows アプリ間で共有できる.dll をビルドします。 く+・ ライブラリ Windows CLR 空のプロジェクト (.NET Core) 開始コアイルを持たぎ、.NET Core をラーゲットとする C++ プロジェクトです。.NET と C++ コード間 の相互運用性を提供します。 C++ ライブラリ Windows
		CLR 空のプロジェクト (.NET Framework) 開始ファイルを持たず、.NET Framework をターゲットとする C++ プロジェクトです。.NET と C++ コード間の相互運用性を提供します。 コンソール C++ Windows CLR コンソール アプリ (.NET Framework) .NET Framework をターゲットとして Windows ターミナルで C++ コードを実行します。.NET と C+ コード間の相互運用性を提供します。

「次へ (N)」のボタンをクリックします。

LR 空のプロジェクト (.NET Framework) コンソール C++ Windows ロジェクト名(N) Project 所し C2¥Users¥sakaw¥source¥repos ・ Jューション名(M) ① *roject1] ソリューションとプロジェクトを同じディレクトリに配置する(D)	R 空のプロジェクト (.NET Framework) コンソール C++ Windows izクト名(b) jesti し Users¥sakaw¥source¥repos • -ジョンス(M) ・ jject1 /リューションとプロジェクトを同じディレクトリに配置する(D) ムワーク(D) T Famework 4.7.2 •	新しいプロジェクトを構成します		
Bジェクト名(Ŋ) Project1 所① C×UJsers¥sakaw¥source¥repos ・ ↓ Jューション名(M) ① *roject1] ソリューションとプロジェクトを同じディレクトリに配置する(D)	/z クト名(L) ject1 Users¥sakaw¥source¥repos ・ -ション名(M) ① ject1 //Jューションとプロジェクトを同じディレクトリに配置する(D) ムワーク(D) T Framework 4.7.2 ・	CLR 空のプロジェクト (.NET Framework) コンソール C++ Windows		
Project1 所(_) C:¥Users¥sakaw¥source¥repos ノューション名(M) ① ?roject1] ソリューションとプロジェクトを同じディレクトリに配置する(D)	yeet ① ① ① ③ yers¥sakaw¥source¥repos	プロジェクト名(1)		
所① Ci¥Users¥sakaw¥source¥repos * Jューション名(M) ① Project1] ソリューションとプロジェクトを同じディレクトリに配置する(D)	① ① FUsers¥sakaw¥source¥repos ● →ションネ(M) ① jject1 ノリューションとプロジェクトを同じディレクトリに配置する(D) ムワーク(F) T Framework 4.7.2 ●	Project1		
C:¥Users¥sakaw¥source¥repos ・ Jューション名(M) ① Project1] ソリューションとプロジェクトを同じディレクトリに配置する(D)	AUsers¥sakaw¥source¥repos ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	易所(上)		
リューション毛(M) ① [?] roject1] ソリューションとプロジェクトを同じディレクトリに配置する(D)	- ション名(M) ① ject1 バリューションとプロジェクトを同じディレクトリに配置する(D) ムワーク(E) T Framework 4.7.2 ・	C:¥Users¥sakaw¥source¥repos •		
Project1] ソリューションとプロジェクトを同じディレクトリに配置する(<u>D</u>)	jject1 /リューションとプロジェクトを同じディレクトリに配置する(<u>D</u>) ムワーク(<u>F</u>) T Framework 4.7.2 ・	/リューション名(M) 1		
〕 ソリューションとプロジェクトを同じディレクトリに配置する(D)	ソリューションとプロジェクトを同じディレクトリに配置する(D) ムワーク(E) T Framework 4.7.2 ・	Project1		
	ムワーク(E) T Framework 4.7.2 ・	ソリューションとプロジェクトを同じディレクトリに配置する(D)		
/-ムワーク(E)	T Framework 4.7.2 •	7レームワーク(<u>F</u>)		
NET Framework 4.7.2 •		.NET Framework 4.7.2		
			戻る(<u>B</u>) 作	成(<u>C</u>)

「場所(L)」の表示が異なっていると思いますが気にしなくてもいいです。どこにプロジェクト(作るプログラム)を保存するか気にしなければ、デフォルトでもいいですが、右側の四角をクリックすると

📢 プロジェクトの場所						×
\leftrightarrow \rightarrow \checkmark \bigstar sakaw \Rightarrow source \Rightarrow repos		~	Ō	reposの検索		P
整理 マ 新しいフォルダー						?
▲ クイック アクセス ■ デスクトップ ★ ■ ダクンロード ★ ■ ドキュメント ★ ■ ビクチャ ★ ■ jyohou ■ jyohou ■ 特徴 ■ 二歩突き切り戦法 Microsoft Visual S ■ repos ● OneDrive ■ PC	更新日時 検索条件に一致する項	種類	<u><u><u></u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u></u>			
フォルダー:						
			7:	ォルダーの選択	キャンセノ	ل ا

とプロジェクトを保存するフォルダを指定できます。私は「C++src」というフォルダを作って、 C++のプログラムはそこに保存するようにしています。

→ * ↑ 🖬 > PC	> Windows (C:) >		ٽ ~			
里 ▼ 新しいフォルダー						- (
- 二歩突き切り戦法 ^	名前	更新日時	種類	サイス	(
Microsoft Visual S	baduk	2020/07/02 11:30	ファイル フォルダー			
repos	C++src	2020/07/11 0:17	ファイル フォルダー			
	FreeSoft	2020/07/08 7:14	ファイル フォルダー			
OneDrive	GAMES	2020/07/09 8:12	ファイル フォルダー			
PC	go	2020/07/01 15:12	ファイル フォルダー			
 ■ 3D オブジェクト	goengine	2020/06/30 19:36	ファイル フォルダー			
	inetpub	2020/05/14 10:25	ファイル フォルダー			
	Intel	2020/07/10 1:47	ファイル フォルダー			
テスクトップ	JUST	2020/05/12 15:26	ファイル フォルダー			
ドキュメント	PerfLogs	2020/05/15 0:53	ファイル フォルダー			
■ ピクチャ	Program Files	2020/07/11 0:07	ファイル フォルダー			
· ビデオ	Program Files (x86)	2020/07/11 0:07	ファイル フォルダー			
b ミュージック	shogi	2020/07/04 21:11	ファイル フォルダー			
G Windows (C)	SWSETUP	2020/05/11 20:34	ファイル フォルダー			
Windows (C.)	texlive	2020/05/12 22:52	ファイル フォルダー			
Recovery Image 🗸	texsrc	2020/06/09 13:43	ファイル フォルダー			
フォルダ	-; C++src					

「フォルダーの選択」のボタンをクリックします。

新しいプロジェクトを構成します CLR 空のプロジェクト(.NET Framework) コンソール C++ Windows プロジェクトを(N) Project1 場所(L) C¥C++src¥ ・ r* YJコージョンを(M) ● Project1 JUコージョンを力切ジェクトを同じディレクトリに配置する(D) フレームワーク(D) INET Framework 4.7.2 ・			-		\times
CLR 空のプロジェクト (.NET Framework) コンソール C++ Windows プロジェクト名(N) Project1 場所(L) C-¥C++src¥ ・ ・ ・・ ソリューション名(M) ① Project1 」 ソリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(D) .NET Framework 4.7.2	新しいプロジェクトを構成します				
プロジェクト名(N) Project1 場所(L) C Y C + + src ¥ ・ ・ ・・ ソリューションを (M) ● Project1 □ ソリューションと プロジェクトを同じディレクトリに配置する(D) フレームワーク(D INET Framework 4.7.2 ・	CLR 空のプロジェクト (.NET Framework) マンソール C++ Windows				
Project1 場所(L) C+YC++srcギ ソリューションズ(M) ① Project1 ソリューションズプロジェクトを同じディレクトリに配置する(D) フレームワーク(D) INET Framework 4.7.2	プロジェクト名(N)				
場所(L) C.¥C++src¥ ・ *** ソリューション名(M) ① Project1 ソリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(D .NET Framework 4.7.2 ・	Project1				
C+¥C++src¥ ・ ソリューションを(M) ① Project1 フリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(E) .NET Framework 4.7.2	場所(L)				
ソリューション名(M) ① Project1 フリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(E) .NET Framework 4.7.2	C:¥C++src¥ v				
Project1 ソリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(E) .NET Framework 4.7.2	ソリューション名(M) 🚺				
□ ソリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(E) .NET Framework 4.7.2 ~	Project1				
フレームワーク(E) .NET Framework 4.7.2 -					
.NET Framework 4.7.2 •	フレームワーク(E)				
	.NET Framework 4.7.2				
		戻る(<u>B</u>)	作成(Q	
戻3(<u>B</u>) 作成(<u>C</u>)					

「プロジェクト名(N)」をプログラムの内容を表す名前に変えることが出来ます。ここでは「windowsProject1」としてみます。プロジェクト名を設定すると自動的に同じ名前が「ソリューション名」に設定されます。「ソリューション名」を別の名前に設定できますが、あえて変える必要はありません。「ソリューション名」と同じ名前のフォルダが「場所」に指定された所に作成され、そのフォルダ内に必要なソフトやフォルダが作られます。

「作成 (C)」をクリックします。

📢 77/1/10 雑葉D 高売扱 70/227HB どん行動 デバク加 デスト会 分析包 アントロ 記録電配位 9/2/F200 ヘルプ10 物面(Cal-Q) 🔑 WindowsProjecti 0 - 0 数・金 論 論 デ ウ - 7 - Debug - 148 · ト D - 7.5 Windows デバッティー 声 図 -		🔁 – 🗗	×
скарта - астат ула	リロン クロン 10 70 日 - 1 0 - 2	Ø ♥ ₽ = (004) (014)	- a x - a
	701-98919370-9- 7- 70/(74 Window Decisit 70/12/04	10/1=/	- 9 ×
	2 24 P		
	 その他 (名前) ブロジェクトファイル ブロジェクトの依存開係 ルート名前空間 	WindowsProject1 C#C++srcWWindowsPro WindowsProject1	iject1¥Wii
±力			
27xxxx • k = k ≡ f ²	(名前) プロジェクト名を指定します。		
		↑ ソース管理に追加 -	- 10

となります。

上段のメニューの「プロジェクト」の「新しい項目の追加」を選択します。

▶ 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)	テスト(S) 分析(N)	ツール(T) 拡張機能(X)	ウィンドウ(W) ヘルプ	(H) 検索 (Ctrl+Q)	Q	WindowsProject1			•	- ø	×
0-0 8-0 8 - 0	↑ ソリューションの再ターゲット	de	ws デバッガー - 🏿 🎜 🚽								A Live Share	8
0 - 0 8 - 2 4 4 4 7		c Crid-Shift-A Shift-Aht-A Shift-Aht-A	ex \$7/(7/7 µ = 60 =						2012-022-02700-2- ○ ○	i d# [9] ★ (Cri+:) sject1'(1/1)	会 Live Share	φ × ρ.
									ソリューション エクスプローラー プロバティ WindowsProject1 プロジェクト・ マーション その他 (名称) プロジェクトの休存間係 ルート名称記聞	しまりスプロ のプロパティ Windows Windows Windows	Project1 WindowsProj	etiWi
出力								* ā X				
出力元(S):	• 2											
									(名前) プロジェクト名を指定します。			
15-一覧 出力						_		 				
[7] 準備完了										↑ ¥-3	御神に追加 -	-

○ペ ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デパッグ(D) テスト	(5) 分析(N) ソール(T) 拡張機能(X) ウ・	インドウ(W) ヘルプ(H) 検索 (Ctrl+Q)	P WindowsProject1		🕒 – 🖉 –
8 ○ • ○ 8 • • ■ ■ • 7 • 7 • Debug • x86 • •	・ローカル Windows デバッガー ・ 声 🗟 🥫				년 Live Share 🛱
					ソリューション 120 200-5 ・ マ × ・ マ × ・ マ ・ マ ・ マ ・ マ ・ マ ・ マ ・ マ ・ マ ・ マ
ž	新しい項目の追加 - WindowsProject1			? ×	▶ •■ 参照 ▶ 前 外部依存着保
uff 927	▲ インストール済み	並べ替え: 既定 🔹 👯 🔚	検索 (Ctrl+E)	- م	≠ ソースファイル → ヘッダー ファイル
	ド ー- ド ー- デ ー- 学 ー- ス リン- ス リン- ス Web	C -= 77/b (App) Ay9-77/b (A) C -= 793	Visual C++ Visual C++ Visual C++	(++	
	ユーディリティ プロパティシート HISL Test グラブパックス				
					ソリューション エクスガローラー チーム エクスガローラー プロパチイ ・ ③ × WindowsProjectt プロジェクトのプロパティ ・ ③ 9 9 / ●
	名前(M): ソースcpp			_	日 その他 (名前) WindowsProject1 プロジェクトファイル C-VC++actWindowsProject1WWi プロジェクトの水界等所。 ルート名前空間 WindowsProject1
	ABITT(L): CIAC++SICAWINDOW	seroject i windowseroject i #	● 2R(0)	3810(A) #47/016	
 出力元(S): 1 と ね 1 	1 1 I I I I I I I I I I I I I I I I I I				<u>^</u>
1 7 1 8 2					(名前) プロジェクトキモ指定します。
□ 準備完了					↑ ソース管理に追加 ▲ 🐴
日 P ここに入力して検索 O 目	- 😑 🔲 🟥 📄 😳	e 👌 者 N			🕐 ^ 🛥 🧟 🕼 A 🗹 2020/07/11 🗣

となります。

新しい項目の追加 - Win	ndowsProject1					?	\times
▲ インストール済み		並べ替え	既定 🔹 🚦			検索 (Ctrl+E)	ρ-
 ✓ Visual C++ コード 書式設定 ŪI CLR データ リソース Web ユーティリティ プロパティシート HLSL Test 			Windows フォーム CLR ユーザー コノトロール	Visual	IC++ IC++	種類: Visual C++ 他の Windows コントロールを含む CLR を作成します	77-6
グラフィックス							
▶ オンライン							
名前(<u>N</u>):	MyForm.h						
場所(<u>L</u>):	C:¥C++src¥Windows	Project1¥	WindowsProject1¥		•	参照(B) 追加(A) キャン	セル

上図のように、「Visual C++」の「UI」の「Windows フォーム」を選択し、「追加(A)」をクリッ クします。

D	3 774/61 編集版 表示的 プロ2257/69 20/510 75/5/10 73745 分析60 7-6/0 営務無数0 75/75/00 AU7/10 常産(04-0 - P WeedowsPreject) ○・○ 13・② 副 14 フ・♡・ Debog ・ x86 - ▶ D-15/Windows 75/75・ 月 図 15 十 二 中 止 回 工会 (日 中 二 回 ス つ) 中 二 が 、		Ive Share R ²
4	Myform為(デザイン) + ×	- 0	ソリューション エクスプローラー
~-バ~ エクスプローラー ツールボックス	Writemal (1967) * * • 7-99% All LSG With SG With SG V.S. 9977-000.30% ANE UK TO IJ- #NK#56.00 % ANE UK • • 1003-100 • • • 2003-100 • •		998-5724 2016-5- 998-5724 2016-5- 998-5724 2017-5-98年(104) 1993-572 2017-5-98年(104) 1993-572 10400-1996 - 1995-592 10400-1996 - 1995-592 - 1
			ソリューション エクスプローラー ゲーム エクスプローラー
			70/(71 ~ 9 ×
			* 101 01 12
	IJ	マ - * # × 15東日秋奈 夕・ 274後 侍	E , * <i>C</i>
	13		
	孝儀充了		↑ ソース管理に追加 🔺 🐴

のようなエラーメッセージが表示されます。上段のメニューの「ファイル」の「ソリューションを 閉じる(T)」を

N	771	ル(F) 編集(E) 表示(V) プロジェクト	(P) ビルド(B)	デパッグ(D) テスト(S) 分析(N) ソール(T) 拡張機能(X) 912ドウ(W) ヘルプ(H) 技法 (Ctrl+Q) の WindowsProject1		🕮 – o ×
		新規作成(N)	,	- ▶ □-カルWindows 約/2/1 - 声 回 : + 日 + 日 戸 + 日 日 王 22 回 井 平 声 % -		ピ Live Share 🔗
1		R((0)	,			
7		74-F047E0W0			-	
Ľ)		ソース管理に追加			$ \uparrow $	00000.0.20000
2H		38±0/0		込み前に以下のエラーを解決する必要があります。		ワリュージョン エジスノローフー の機業 (C01+1) ア・ マ アリコージョン 20(adama Depinent) (11) プロジェクト)
4		III (MC)				S WindowsProject1
4	តា	1月しか(C) ソリューションを閉じる(T)				▲ +# 泰照
		Live Share セッシュンを開始		Exit (5-09939-ax800000A)		msconio System
252		Live Share セッションに参加				System.Data
		McCorre h (h G200/S)	ChileS			System.Drawing
		名前を付けて MyForm.h を保存(A)				 System.Xml
	а.	すべて保存(L)	Ctrl+Shift+S	Windows フォームのデザイン時エラー一覧のチェック		▶ 💼 外部依存関係
		ソース管理(R)	,			▲ 🔛 ソース 7711// ** MyForm.cop
		ページ設定(U)				4 🖾 No9-7711
		印刷(P)	Ctrl+P			Myforn.h
		アカウントの設定()				
		最近使ったファイル(F)	,			
		最近使ったプロジェクトとソリューション(J)	•			
		MyForm.h の移動先(I)	•			
		終了(X)	Alt+F4			
	15 903	5度 →2023年 ● ● 025- ● 】 1 3-F 世界	<u>k</u> 078 tr 0 0	(137gでジ 译 ビルド・MatiSona ・ 15111日間 プロジェジト ファイル 行	∨ • م •	992-942 1930-5- 740fri - 0 x 20]91 ≠ 20]91 ≠
0	偏完	7				↑ ソース策環に追加 🔺 🍖
	2	○ ここに入力して検索		o 🛱 🤮 📠 🏛 💿 🧔 🙋 🖪 🕍 🚺		🕜 ^ 🛥 🖟 (1) A 🖸 12:34 🛃

クリックします。

D① ファイルKF) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デパッグ(D) ※ ○ + ○ 記 + 🎥 🔐 ♡ + ♡ + □ Debug + x85	テスト(5) 分析(N) ソール(T) 拡張機能(X) ウィンドウ(W) ヘルブ(H) ・ ▶ ローカル Windows デバッガー ・ ^{pl} 図 _マ	検索 (Ctrl+Q)	٩	🕚 – O × Lê Live Share 🖉
2 274/0	23.45 5 560 2-2-27 0 33年80 2-27200 3-3791 ▶ 0-25 Wordson 9757 」 ■ 回。 実行する操作を選択してくた 最近期いた項目(R) ■近位世界(A)	19日 IICM-CD ださい 2003/07/11 11:55 2003/07/11 0:25 2003/07/10 1:69 2003/07/10 1:69 2003/07/10 1:69	 	ि – ० × Rituesteer R ००० £ स + ७०६ व ७ । ० ⊁ –
19	Chearcr道道君子道中道品KardGarde《Wahittet	2020/07/09 7.29	プロジェクト ファイル (7 原制状態	 • • х <i>P</i> - У№->02.127.00-3- 3-1.127.70-3-
 ロ 単 2ごに入力して検索 O 	H 😄 🚍 💼 💼 🧿 🖉 🖪 🖆			40 () ∧ ← ≪ 48 A [] 1242 2020/07/11 🕎

となります。左側の「最近開いた項目 (R)」の「今日」の一番上にプロジェクト名の「WindowsProject1.sln」がありますから、それをクリックします。

X] アイルの 範囲の 高売10、プロジェアトの 22人内の 81大0/10 81大0/10 81大0/ 72人名 分析00 ソールの 起意発展の クロゲン(M) ヘルプル 10 第三(ローロ)	6) − □ Live Share	×
		The second s	-	-
7	Myformh(7)/1/) + X · · · ·	ソリューション エクスプローラー		- ¶ ×
-パー エクスプローラー ツールボックス		2012-0000000000000000000000000000000		ρ.
		ソリューション エクスプローラー <u></u> チーム エク	770-5-	
		70/(71		≁ q ×
		MyForm System Windows Forms Fo	orm	
		웬 말 안 두 온		
		StartPosition Wine	dowsDefaultLocation	1 ^
		Windowstate Non	mai	-
		BackColor	Control	
		BackgroundImage	(なし)	
		BackgroundImageLayout Tile		
	15	Cursor Defa	ult	- 1
	V/1Va/2dz • 1015- 4.035- 0.0378-9 1/6 + IntelSence • 1035880	ForeColor	ControlText	-
		FormBorderStyle Siza	ble	
	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	RightToLeft No		
		RightToLeftLayout False		- 1
		lext MyF	orm	~
		Text		
		コントロールに関連付けられたテキストです		
	神徳??		ソース管理に追加 🔺	

となります。昔の visual studio C++ はこの状態でビルドするとよかったのですが、「ビルド」の「ソリューションのビルド(B)」をクリックすると

	- 0	×
20 - 0 (0 - 2 - 0) X - 1 (0 - 0) X	EP Live share	×
Mytema 1970(2) = × • 102-102 (21-2) Wytema 1970(2) = × • 0 202-102 (21-2) Wytema 1970(2) = × • 0 202-102 (21-2) Wytema 1970(2) = × • 0 202-102 (21-2) Wytema 1000 • 0 0 0 • 0 0 0 0 0 • 0 0 0 0 0 0 • 0 0 0 0 0 0 0 • 0 0 0 0 <t< th=""><th>Ф 🖗 О 🖋 – – 1 Сон-3 екст (Ил / ФУ29) - Seforme</th><th>- 4 ×</th></t<>	Ф 🖗 О 🖋 – – 1 Сон-3 екст (Ил / ФУ29) - Seforme	- 4 ×
	19370-9-	
20.74 20.74		• 4 X
	ingin of the	-
tes ar Carlos (Constanting) Santositate (Constanting)	WindowsDefaultLocatio Normal	in *
E RA Baccor Baccor Bactor Baco	Control (なし) Tile	
15-−≅ Cutor Cutor C	MS UI Gothic, 9pt	-
ソリューション会体 ・ ②112- ▲0巻巻 ① 0.5/2セジ 学 ビルド + InteliSence ・ 23 夏を快会 P ForeColor	ControlText	
コード 説明 プロジェクト ファイル 行 別の秋秋巻 FormStanderSyle S	Sizable	
C LNK1501 エントリーボイント在空観しなければなりません。 WindowsProject1 LNK 1 Bightfoldet Pro	NO False	- 1
Text	MyForm	
Text 32/10-16環境的的た状況 32	F.C.F.	
	↑ ソース管理に追加 →	-

のようにエラーメッセージが表示されます。このフォームを実行時に作成して表示するためのプロ グラムを用意する必要があります。

上段のメニューの「プロジェクト」の「新しい項目の追加」を選択します。

新しい項目の追加 - Wir	ndowsProject1				?	×
▲ インストール済み		並べ替え: 既定 ▼ 詳 ☷		検索 (Ctrl+E)		ρ-
✓ Visual C++ コード 書式設定 UI CLR データ リソース Web ユーティリティ プロ/ディシート HLSL Test グラフィックス		 C++ ファイル (.cpp) 	Visual C++ Visual C++ Visual C++	種類: Visual C	**	
▶ オンライン						
名前(<u>N</u>):	ソース.cpp					
場所(<u>L</u>):	C:¥C++src¥Windows	Project1¥WindowsProject1¥	~	参照(<u>B</u>)	追加(<u>A</u>) キャン・	セル

の画面で、今度は、「Visual C++」の「コード」の「C++ファイル (.cpp)」を選択し、

新しい項目の追加 - Wir	ndowsProject1				?	\times
▲ インストール済み	1	並べ替え: 既定 🔹 👬 📘		検索 (Ctrl+E)		ρ.
 ✓ Visual C+++ □=F ⊕ # 3 t ⊕ # UI CLR データ リソース Web ユーデオ(リティ) Taxif(ディシート) HLSL Test グラフィックス ▶ オンライン 		C++ ファイル (.cpp) ハッグ-ファイル (.h) ハロレファイル (.idl) コンポーネント クラス モジュール定義ファイル (.def) CLR ユーザー コントロール C++ クラス	Visual C++ Visual C++ Visual C++ Visual C++ Visual C++ Visual C++	種類: Visual C++ 1203		
名前(<u>N</u>): 場所(<u>L</u>):	ソース.cpp C:¥C++src¥WindowsP	roject1¥WindowsProject1¥	•	参照(B)	+ 11	7 11.
				12/11(<u>A</u>)	1471	-72

「追加(A)」をクリックします。

📢 ファイル(D) 編集(E) 表示(U) プロジェクト(D) ビルド(B) デバック(D) テス	NS 分析(N) ツール(D) 拡張機能(X) ウインドウ(M) ヘルプ(H) 検索(Ctrl+Q)	P WindowsProject1		🛑 – a ×
🕴 🖸 - 🔘 🛐 - 🏠 🔛 🔐 🔊 - 🤆 - 🛛 Debug - 🕺 x86 - 🛛	• ローカル Windows デバッガー • 🏓 📾 🚎 🖿 🌾 🗍 🕲 👘 🗍 🖉 🗍 📮			ഥ Live Share 🖉
学 <mark>ソースcpp + ×</mark> MyForm.h (デザイン)			- 0	ソリューション エクスプローラー
10% •	 (ワレーCD-23-7) ビンド・IntelSense ・ 	• 70/221- WindowPojet1 LINK	・ ① ・ ① ・ ① ・ ① ・ ① ・ ① ・ ① ・ ② ・ ② ・ ② ・ ② ・ ② ・ ③ ・ ③ ・ ③ ・ ③ ・ ③ ・ ③ ・ ③ ・ ③	y12-592 12270-5- ● ● 3 ● ● ● ● ● ● ● ● ● ● ● ● ● y13-592 12270-5-0000 (0+0) ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
15				
□ 神儀充了				↑ ソース管理に追加 🔺 🐴

と「ソース.cpp」が作られ、何も書いてない空のファイルが表示されます。



のように

#include "MyForm.h"
using namespace WindowsProject1;
[STAThreadAttribute]
int main() {
 MyForm[^] fm = gcnew MyForm();
 fm->ShowDialog();
 return 0;
}
と打ち込みます。ここで、
using namespace WindowsProject1;

の WindowsProject1 はプロジェクト名です。 「ビルド」の「ソリューションのビルド(B)」をクリックすると

×	】 ファイルビ 編集(E) 表示(M) プロジェクト(E) どルド(B) デパック(D) ◎ • ◎ 13 • 🖕 🔐 🄊 • 🤆 • Debug • x86	テストS 分析(M) ツール① 拡張機能(M) ウインドウ(M) ヘルプ(H) - ▶ ローカル Windows デバッカー - 声 回 - h (作 当 注 ■ *	技策(Ctri+Q) ア WindowsProject1	Ive State	×
サーバー エクスプローラー・ツールボックス	VPS200 + K (Micros)(FX) VMConsuper- S under Research (Micros)(FX) S under Research (Micros)(FX) (STATuredd(triate) (STATuredd(triate)) (STATuredd(tria	- (90-10,33-7)	• 0 main)	P 2/25-2022 25/25-2- P 2/25-2022 25/25-2- P 2/25-2023 25/25-2- P 2/25-2023 25/25-2- P 2/25-2023 25/25-2- P 2/25-2023 25/25-2- P 2/25-2025 25/	- α × ρ -
				ソリューション 20スプローチョー プロパティ 国際 日本 (多)	* 7 ×
	10% 0 数単な見つかりおんてした 立方 立方の回してなど した。一部 した。一部 に、した。可能は、プロジェンクト1: NickowProject1, 単元に、Pake Payment 4: NickowProject1, 単元の、Payment 4: UK-Payment 4: UK-Pa	Not		- 6:10 2;R2 p7 ctar - 0:2 X	
	17-一覧 <u>出力</u> ビルF正常終了				-{0

のように、正常にビルドできます。

「デバッグ」の「デバッグの開始(S)」をクリックすると

0 ¹ 77(K) 電販店 巻きが 705/25/P だんRB デバップ(0) 72/KB 学校の 9-K/T 医装置数の 90/25/W ヘルプルP 地生 Col+Q ア ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	\oplus – σ $ imes$ Live Share \vec{R}'
	ビアント ・ (× Y) ・ ・ (× Y) ・ (× Y) ・ ・ (× Y) ・ ・ (× Y) ・ ・ (× Y) ・ ・ ・
100% - の無難は見つかりませんでした	
	+ + ×
【京((Lint) シ ・マーク機能の理想】 ************************************	8 De ~
	A 11-76221-764
	(?) ∧ ♠ /@ 41 © U 13:39 2020/07/11 ₹2

のように、フォームが表示されます。黒い画面も一緒に表示されますが気にしなくてもいいです。 随分、手間がかかりましたが、何もないフォームを表示することが出来ました。

これまでの手続きは魔法使いの儀式だと思って、以下のように、フォームに「ツー ルボックス」にあるツールを配置して、プログラミングをすればいいです。

後は以下のようにプログラミングすれば良いです。VC++ 2010 と VC++ 2013 と VC++ 2019 に共通です。

ツールボックスを表示し、Form に PictureBox と TextBox 1個と Button 2個とを配置する。

00		sudoku - Microsoft Visual Studio Academic	- 8 ×
771.	ル(F) 編算	&(E)表示(V) Qt5 プロジェクト(P) ビルド(B) デバッヴ(D) チーム(M) データ(A) 書式(O) ツール(T) テスト(S) ウィンドウ(W) ヘルプ(H)	
1 60	- 🔛 - ⊿	🌶 🔜 🥔 🐇 🖧 🖏 👘 🔹 🖓 🔸 🔍 🕨 Debug 🔹 Win32 🔹 🧖 💩 🔹 🔹 👘	
主導	12 4	■ ●●■ 回	
a a	Form1.h [デザイン]* × レリューション エクスプローラー クラス ビュー	▼ ツールボックス ▼ ♀ ×
4	17-000000000000000000000000000000000000		▲ ⇒ すべての Windows フォーム ^
÷.	(<u></u>	Formi	▲ コモン コントロール
5			ト ボインター
겁			ab Button
1			CheckBox
T.			R CheckedListBox
			ComboBox
			DatelimePicker
		Button I	A Label
			222 LietView
			22 ListView
		button2	B- MaskedTextBox
			MonthCalendar
			NotifyIcon
			NumericUpDown
			PictureBox
			m ProgressBar
			 RadioButton
1	出力		RichTextBox
	出力元の表	示(5):	webl TextBox
			ToolTip
			t TreeView
			R WebBrowser
			▲コンテナー
			R ホインター
			Se N-1 #wb7
		· // · · ·	X 7 10097A B 20/04
進備3	e r	115,15	<u>∓</u> [746 x 427

更に、MenuStrip をクリックし、Form1 上でクリックする。

sudoku - Microsoft Visual Studio Academic	- 🗆 ×
ル(E) 編集(E) 表示(V) Q15 プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) ツール(I) テスト(S) ウィンドウ(W) ヘルプ(H)	
* 🗃 * 📸 😹 🖗 * 🖓 * 🖓 * 🖓 * 🔍 🕨 Debug 🔹 🛛 Win32 🔹 👘 🖓 abs 🔹 🔹 💎 🖓 🐨 🖓 絶 塑 コ * 🖕	
Form1.h [デザイン]* × ソリューション エクスプローラー	→ ツールボックス → 年 >
	▲ TreeView
- Formi	WebBrowser
27-32#	*コンテナー
	▶ ポインター
	FlowLayoutPanel
	GroupBox
	ColitContainor
butten 1	TabControl
	Tablel avoutPapel
	▲ メニューとツール バー
hutten?	ト ポインター
Denote .	ContextMenuStrip
	MenuStrip
	Y 🔚 StatusStrip
0	ToolStrip
W menuStrip1	ToolStripContainer
	▲ T-9
	ト ホインター
出力	→
出力元の表示(5):	S ^a BindingNavigator
	BindingSource
	- DataSet
	4 コンボーネント
	ト ボインター
	BackgroundWorker
	シールボックス 🚰 プロパティ
₩7	

メニューの「ここへ入力」をクリックし、順次「Sample」、「Sample1」、「Sample2」と入力する。

sudoku - Microsoft Visual Studio Academic	- D ×
ファイル(E) 補集(E) 表示(Y) Qt5 プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) 書式(O) ツール(I) テスト(S) ウインドウ(W) ヘルプ(H)	
🗄 🖥 * 🖼 * 😰 👦 🕼 🖇 🖧 🖄 - P. P 🔍 - 🔍 - 🔍 - 🔍 Debug 🔹 Win32 🔹 💋 abs 🔹 🔹 🖓 😤 🔐 🖸 * 🖕	
[確臣を計算の理己が解離]の突然ぬ」を乾薪薪(田田)[[4]4][[2]]。[阿臣計算理][[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
	N 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
t t t t t t t t t t t t t t t t t t t	
	WabBrowcor
Form1	A TYFT-
Sample ZZAX	ト ボインター
3 Sample1	Elowi avoutPanel
Sample2 ZZAJJ	GroupBox
22^\\\\\	Panel
	SplitContainer
button 1	TabControl
	TableLayoutPanel
	メニューとツール バー
button2	ト ポインター
	ContextMenuStrip
	🗟 MenuStrip
	Y L StatusStrip
120	0 ToolStrip
He menuStrip1	ToolStripContainer
	▲ 9 ⁻⁹
	ト ポインター
出力	- # × 🛄 Chart
出力元の表示(5): ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	BindingNavigator
	BindingSource
	DataGridView
	DataSet
	*
	RackgroundWorker
	SA V-ILITWAZ
267.7	<u>∓</u> []32 × 19

プロパティを表示し、TextBox の Text を 1 とし、 Button の Text をそれぞれ「数値入力」と「Hint」にし、適当なフォントを選び、PictureBox のサイズを 360,360 にする。

00	sudoku -	- Microsoft Visual Studio Academic		- 0 ×
ファイル(F) 編集(E) 表示(V) Qt5 プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) テータ(A) 書式(O) ツール(T) テスト(S) ウィンドウ(W) ヘルプ(H)		
i 🔂 • 🖽 • 😂 属 🥔	🕉 🖧 🖄 🍠 🔹 🖓 - 🖓 - 🖾 🕨 Debug 🔹 Win32	🔹 🤷 abs 🔹 🔹 🖓 🖓 🞲 🥸 💥 🛃 💶 📲		
御 臣 今 司 東 ゆ	표 글 한 冠 尊 ~~ 말 밖 밖 하 하 다 된 것 것 물	비랑, [20] [6 비 7 4] 코 10] [H 또 [급 11 28] [[] 다 .		
Form1.h [デザイン]* ×	ソリューション エクスプローラー クラス ビュー		→ プロパティ	+ # ×
4			 Form1 Sys 	stem.Windows.Form +
7	Form1		81: 24 U	1 / 12
Sample			Context	Men (なし) ^
8			DoubleF	Juffe False
4			Enabled	J True
			ImeMor	de NoControl
			- 123K	
		White a st	AutoSca	aleMc Font
		效值入力	AutoScr	roll False
			AutoScr	oliMi 0, 0
			> AutoScr	OIMI 0, 0
		Hint	AutoSiz	e Faise
			> Locatio	n 0.0
			b Maximi	mSI: 0, 0
		•	Minimu:	mSiz 0, 0
Diaman Christian			Padding	0, 0, 0, 0
imenustrip1			> Size	746, 466
			StartPo	sitior WindowsDefau
			Window	/Stat Normal
出力			▼ ♥ × ▲ 表示	
出力元の表示(S):	· 9 9 9 3 3 9		BackCol	ior Control
			Backgro	
			Backgro	Jundi Tile
			E Eont	MS UI Cothic
			ForeCol	or ControlTe
			FormBo	order Sizable
			17 m	
			× 9-11.759	
-281(e)70 J			-141 I./	40 X 400

メニューの Sample1 をダブルクリックする。

sudoku - Microsoft Visual Studio Academic	- 0	*
かイル(E) 編集(E) 表示(V) Qt5 プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) ツール(I) テスト(S) ウィンドウ(W) ヘルプ(H)		
🛐 + 🖼 + 😰 🚽 🕼 🖕 🛵 🖄 + 🖓 - 🔍 - 💭 + 🖳 🕨 Debug 🔹 🛛 Win32 🔹 💆 💩 bs 🔹 🔹 🖓 🕾 💥 💭 😓 💆 🗆 +		
◎ 4.5.4 作用 第第一回 2.4 回 2.4 回 1.4 目 7.4 回 回		
Form1.h × Form1.h (デザイン)* ソリューション エクスプローラー クラス ビュー		×
* sudoku::Form1 - @sample1ToolStrinMenuItem_Click(System::Object_A_sender_System::FventArgs_A_e)	sample1ToolStripMenuItem	
<pre>this>lame : L Form1: this>lame : L Form1: this>menuStripl>Performacut(false): this>menuStripl>Performacut(); (c]::safe_cast(System::DomcontModel::ISupportInitialize^ >(this>pictureBox]))>EndInit(); this>menusulf); for anno endresion private: System::Void sampleToolStripMenuItem_Click(System::Object^ sender, System::EventArss^ e) []];</pre>	Converse samples Tools private Converse False File Converse False Tools Tools	Stri
	-	- 1
	※ ツールボックス 雪 カルディ	
4億元了 107 月 107 月	4) 104 文字 #	え

private: System::Void sample1ToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
}

を

private: System::Void sample1ToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {

```
とする。
```



さらに、この上に(使う前に ShowBan()を定義しなければならないので)

```
private: void ShowBan() {
  Graphics^ g = pictureBox1->CreateGraphics();
  Brush^ brush = gcnew SolidBrush(Color::White);
  g->FillRectangle(brush, 0, 0, pictureBox1->Width, pictureBox1->Height);
  Pen^ pen = gcnew Pen(Color::Black, 1);
  int K = 9;
  int s = pictureBox1->Height / (K+2);
  int w = (pictureBox1->Width - s * 9) / 2;
  int h = (pictureBox1 -> Height - s * 9) / 2;
  for (int i=0; i<=9; i++) {</pre>
    if (i % 3 == 0) {
      pen = gcnew Pen(Color::Black, 3);
      g->DrawLine(pen, w, h+i*s, w+9*s, h+i*s);
    } else {
      pen = gcnew Pen(Color::Black, 1);
      g->DrawLine(pen, w, h+i*s, w+9*s, h+i*s);
    }
```

```
}
for (int i=0; i<=9; i++) {</pre>
 if (i % 3 == 0) {
     pen = gcnew Pen(Color::Black, 3);
     g->DrawLine(pen, w+i*s, h, w+i*s, h+9*s);
 } else {
     pen = gcnew Pen(Color::Black, 1);
     g->DrawLine(pen, w+i*s, h, w+i*s, h+9*s);
 }
}
for (int i=0; i<9; i++) {</pre>
 for (int k=0; k<9; k++) {
    if (ban[i][k] > 0) {
      System::Drawing::Font^ drawFont = gcnew System::Drawing::Font("Arial", 24);
      Brush^ drawBrush = gcnew SolidBrush(Color::Black);
      System::String^ drawString = System::Convert::ToString(ban[i][k]);
      g->DrawString(drawString, drawFont, drawBrush, (float)(w+i*s), (float)(h+k*s));
    } else if (ban[i][k] == 0 && ans[i][k] > 0) {
      System::Drawing::Font^ drawFont = gcnew System::Drawing::Font("Arial", 24);
      Brush^ drawBrush = gcnew SolidBrush(Color::Red);
      System::String^ drawString = System::Convert::ToString(ans[i][k]);
      g->DrawString(drawString, drawFont, drawBrush, (float)(w+i*s), (float)(h+k*s));
    }
 }
7
if (HintP) {
 for (int i=0; i<9; i++) {</pre>
    for (int k=0; k<9; k++) {</pre>
      if (ans[i][k] == 0) {
        System::Drawing::Font^ drawFont = gcnew System::Drawing::Font("Arial", 9);
        Brush^ drawBrush = gcnew SolidBrush(Color::Blue);
        for (int j=0; j<n_cand[i][k]; j++) {</pre>
           System::String^ drawString = System::Convert::ToString(cand[i][k][j]);
           if (j < 3)
             g->DrawString(drawString, drawFont, drawBrush,
                 (float)(w+(i+0.2*j)*s), (float)(h+k*s));
           else if (j < 6)
             g->DrawString(drawString, drawFont, drawBrush,
                 (float)(w+(i+0.2*(j-3))*s), (float) (h+(k+0.3)*s));
           else
             g->DrawString(drawString, drawFont, drawBrush,
                 (float)(w+(i+0.2*(j-6))*s), (float)(h+(k+0.6)*s));
        }
```

```
}
}
mouseDownP = false;
mouseX = mouseY = -1;
}
```

を打ち込む。

00	sudoku - Microsoft Visual Studio Academic		- • ×
ファイノ	ν(Ε) 編集(Ε) 表示(V) Qt5 プロジェクト(Ε) ビルド(Β) デパッグ(D) チーム(Μ) テータ(Δ) ツール(Ι) テスト(S) ウインドウ(W) ヘルプ(H)		
1 60	- 🖬 • 🧭 🚽 🎉 🎝 🖄 🖄 🕫 - 😂 - 🏭 - Debug - Win32 - 🔹 🧭 abs - 🔹 🖓 🖓 📸 💥 🛃 🔜		
	≤≤⇒≤==================================		
11 F	omi.h* × Form.h (デザイン)* シリューション エタスプローラー クラス ビュー	- プロパティ	+ # ×
4	tsudoku::Form1 - @sample1ToolStripMenuItem_Click(System::Object ^ sender, System::EventAras ^ e)	 sample1To 	olStripMenuItem_ •
-/(- 1/3,70-5-	<pre>standbuilter under version versio</pre>	Access CanOver File FullNam IaConte IaSabete IaShare IaShare IaShare IaShare IaShare VyeStr V	sample 1 ToolStr private de False critosarce 4 Millio critosarce 4 Millio de False False False False False False False False False False False False False
10.00	7	× У−л.#у	双 濟 九/5ィ

```
Form1.h の先頭の
```

```
namespace sudoku {
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
```

の後ろに

```
int ban[9][9];
int ans[9][9];
int candYoko[9][9][10];
int n_candYoko[9][9];
int candTate[9][9][10];
int n_candTate[9][9];
int candGroup[9][9][10];
int n_candGroup[9][9];
int cand[9][9][10];
int n_cand[9][9];
bool HintP;
```

```
bool mouseDownP;
int mouseX, mouseY;
```

と打ち込む。

sudoku - Microsoft Visual Studio Academic		- 0 ×
ファイル(E) 編集(E) 表示(Y) Qt5 プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) ツール(II) テスト(S) ウィンドウ(W) ヘルプ(H)		
🔚 📲 📲 🖉 😹 🕼 👘 🗧 🕾 👘 🖓 🔹 🖓 🔹 🖓 🔹 🖉 👘 🔛 🕨 Debug 🔹 Win32 🔹 🖉 🖓 abs 🔹 🖓 🖓 🖓 🖓 🖓 🖗 🖑 🖉 👘		
[\$P\$\$P\$ ★ \$P\$] 得得一定的。 \$P\$ \$P\$ \$P\$ \$P\$ \$P\$ \$P\$ \$P\$ \$P\$ \$P\$ \$P		
■ Form1.h 「キザイン」* ソリューション エクスプローラー クラス ピュー	プロパティ	+ # ×
	HintP VCCode	ariable •
Constant and the second s	Rea ALLINE	
Using namespace System: Windows: Forms:		
using namespace System::Data;	4 C++	111-110
using namespace System::Drawing;	(Name)	CiVtoyce/Mater
int her [0][0]	FullName	sudoku::HintP
int arc[9][9]:	InitExpressi	
int candYoko[9][9][10];	IsConstant	False
int n_candYoko[9][9];	IsInjected	False
int candiate[9][9][10];	IsShared	False
int n_candiate[3][3][10]:	IsVolatile	False
int candidate [919]	TypeString	bool
int cand[9][9][10];		
int n_cand[9][9];		
bool Hintr's	1	
int moleculum in the second seco		
100% • <		
出力 ママン		
出力元の表示(5):		
	1	
	AD M H Hub7	
	X. 9-June92	m. Junya

実行する。

-	Form1	- 🗆 🗙
Sample		
		0
		数值入力
		Hint

メニューの「Sample1」をクリックする。

	Form1	- C ×
Sample		
	1 9 3	
	9 7	数值入力
	7 1	
	6 9	Hint
	85394	
	<u></u>	

Form1.h[デザイン]をクリックし、「Hint」という名前にしたボタンをダブルクリックする。



private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
 }

を

```
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    if (HintP) {
        HintP = false;
    } else {
        getCand();
        HintP = true;
    }
    ShowBan();
```



```
とする。
```

```
        Image: constraint of the stand of
```

更に、この上に

```
private: void getCand() {
    // cand[][][], candYoko[][][], candTate[][][], candGroup[][][]の初期化
    for (int i=0; i<9; i++) {</pre>
      for (int k=0; k<9; k++) {</pre>
            n_cand[i][k] = n_candYoko[i][k]
             = n_candTate[i][k] = n_candGroup[i][k] = 0;
        }
    }
    // ans[][] から candYoko[][][] のデータを設定
    for (int k=0; k<9; k++) {</pre>
        for (int i=0; i<9; i++) {</pre>
             if (ans[i][k] > 0) {
                 candYoko[i][k][0] = ans[i][k];
                 n_candYoko[i][k] = 1;
                 continue;
            }
             for (int n=1; n<=9; n++) {</pre>
                 bool flag = false;
                 for (int j=0; j<9; j++) {</pre>
                     if (ans[j][k] == n) {
                         flag = true;
                         break;
                     }
                 }
                 if (!flag) {
                     candYoko[i][k][n_candYoko[i][k]++] = n;
                 }
            }
        }
    }
    // ans[][] から candTate[][][] のデータを設定
    for (int i=0; i<9; i++) {</pre>
        for (int k=0; k<9; k++) {</pre>
             if (ans[i][k] > 0) {
                 candTate[i][k][0] = ans[i][k];
                 n_candTate[i][k] = 1;
                 continue;
            }
             for (int n=1; n<=9; n++) {</pre>
                 bool flag = false;
                 for (int j=0; j<9; j++) {</pre>
                     if (ans[i][j] == n) {
                         flag = true;
```

```
break;
                  }
             }
             if (!flag) {
                 candTate[i][k][n_candTate[i][k]++] = n;
             }
        }
    }
}
// ans[][] から candGroup[][][] のデータを設定
for (int t=0; t<3; t++) {</pre>
    for (int y=0; y<3; y++) {</pre>
        for (int k=3*t; k<3*(t+1); k++) {</pre>
             for (int i=3*y; i<3*(y+1); i++) {</pre>
                 if (ans[i][k] > 0) {
                      candGroup[i][k][0] = ans[i][k];
                      n_candGroup[i][k] = 1;
                      continue;
                 }
                 for (int n=1; n<=9; n++) {</pre>
                      bool flag = false;
                      for (int k2=3*t; k2<3*(t+1); k2++)</pre>
                          for (int i2=3*y; i2<3*(y+1); i2++) {</pre>
                               if (ans[i2][k2] == n) {
                                   flag = true;
                                   break;
                               }
                          }
                          if (!flag) {
                               candGroup[i][k][n_candGroup[i][k]++] = n;
                          }
                 }
            }
        }
    }
}
// get cand[][][] = candYoko[][][] & candTate[][][] & candGroup[][][]
for (int k=0; k<9; k++) {</pre>
    for (int i=0; i<9; i++) {</pre>
        bool flag;
        for (int j=0; j<n_candYoko[i][k]; j++) {</pre>
             flag = false;
             for (int t=0; t<n_candTate[i][k]; t++) {</pre>
```
```
if (candYoko[i][k][j] == candTate[i][k][t]) {
                         flag = true;
                         break;
                     }
                 }
                 if (!flag)
                     continue;
                flag = false;
                 for (int g=0; g<n_candGroup[i][k]; g++) {</pre>
                     if (candYoko[i][k][j] == candGroup[i][k][g]) {
                         flag = true;
                         break;
                     }
                 }
                 if (flag) {
                     cand[i][k][n_cand[i][k]++] = candYoko[i][k][j];
                 }
            }
        }
    }
}
```

と打ち込む。



実行し、メニューの「Sample1」をクリックし、「Hint」のボタンをクリックする。

	Form1	- 🗆 🗙
Sample		
		1
	$\frac{457}{8}$ 1 $\frac{256}{8}$ $\frac{24}{8}$ $\frac{456}{8}$ $\frac{246}{9}$ 9 $\frac{245}{68}$ 3	
	³⁴⁵ 9 ²³⁵ ⁴²³ ¹⁴⁵ 7 ²⁴⁵ ²⁴⁵ ²⁵⁶	数值入力
	$\frac{3^{45}}{8^{8}}$ $\frac{2^{45}}{6^{8}}$ $\frac{2^{35}}{6^{3}}$ $\frac{2^{34}}{9^{8}}$ $\frac{4^{56}}{6^{9}}$ $\frac{2^{34}}{6^{8}}$ $\frac{2^{45}}{7}$ 1	
	6 458 9 47 147 123 123 123 257 458 8	Hint
	¹³⁴ ⁴⁵⁸ 7 6 ¹⁴⁹ ¹²³ ¹²³ ¹²³ ¹²³ ⁹	
	2 ⁴ ¹³ 8 ¹⁴⁷ 5 ¹³⁴ ¹³⁴ ⁶⁷⁹	
	157 2 50 4 179 197 8 567 56 75 157 a 156 147 b 146 156 156 567	
	9 ⁶⁷ 3 6 ⁶ 9 ⁷⁷ 2 9 ⁷⁸ 8 ⁶ 8 ⁶⁷ 17 267 0 5 0 16 126 0 4	
	8 5 3 7 9 4	

Form1.h[デザイン] をクリックし、PictureBox をクリックし、プロパティの「イベント」を表示し、

🛥 sudoku - Microsoft Visual Studio Academic	- 8 ×
ファイル(F) 編集(E) 表示(V) Qt5 プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) 書式(O) ツール(T) テスト(S) ウインドウ(W) ヘルプ(H)	
i 🔂 • 🖼 • 📁 🖉 🛃 🍻 👗 🖄 - 🕾 - 🔍 • 🔍 - 🕨 Debug - 🛛 Win32 - 🗸 💋 abs - 🕘 - 🖓	😁 🖬 🕸 🎠 🛃 🔜 * -
	1 (1) (93
🏢 ソリューション エクスプローリー クリス ビュー Form1.h [デザイン] 🗙	 プロパティ → # ×
	∧ pictureBox1 System.Windows •
Form1	25 41 10 9 10
5 Sample	Click
g	DoubleClick
	MouseCaptu
	MouseClick
	MouseDoubl
教値入力	a → DraviewKey/
and of the second se	1 7 -4
	DataBinding
Hint	ドラッグ アンド ドロップ
	DragDrop
9 P	DragEnter
	DragLeave
	GiveEeedbar
	QueryContin
Simeri Strin1	▲ フォーカス
	Validated
	Validating
and a	
	Background
	AUC #SETUCE #D #1 15
audoku.exe (マネーン(+4.0.3031)): 'C:WINDOXAMIOroad'Line(+assamb)が40- NSLLVsystemV4.0.4.0.0_b7365618346083Vsystem24.0.11 が読み込ま	nation BindingConte
sudoku.exe* (マネージ (v4.0.30919)): C:WINDUSSWIICropott.NetVassenblyWAU_HSILVAccess/biittyv4.0.4.0.0.0_b0915f711d90834Vaccess/biitty.d (sudoku.exe* (マネージ (v4.0.30918)): C:WINDUSSWIICropott.NetVassenblyWAU_HSILVAccess/biittyv4.0.4.0.0.0 ja b77455813340895Waccerib	II か焼み込まれました resources.dll が焼み込まれました ClientSizeCh
ブログラム [4852] sudoku.exe: マネージ (v4.0.30319) [オコード 0 (0x0) で終了しました。	ContextMen v
	> 🔆 ツールボックス 😁 プロパティ
	1061 , 44 ≆Ö360 × 360

「マウス」の MouseDown をダブルクリックする。

0	udoku - Microsoft Visual Studio Academic			
ファイル(E) 編集(E) 表示(V) Qt5 プロジェクト(E) ビルド(B) デバッグ(D) チーム(M) データ(A	ツール(工) テスト(S) ウィンドウ(W) ヘルプ(日)			
i 🛅 • 🛅 • 😂 🛃 🥔 🕺 🚵 🖏 🔊 • 🔍 • 💭 • 🖏 🕨 Debug 🔹 Win32	- 🧭 abs	- I 🔍 🕾 🖓 🞲 🕺 🎾 🗉 - 🖕		
[김원님 사업] 建建 [일일] 그 위의 위의 원을 짓 [12] [6 비 개	표 [편 166])에 꽃 [큐 슈] 128 111 부 -			
アリューション エクスプローラー クラス ビュー Form1.h* × Form1.h 「デザイン]*			→ プロパティ	+ # ×
4 Asudoku::Form1	 PrictureBox1_MouseDown(System::Obj 	ect ^ sender, System::Windows::Forms::Mo	pictureBox	1 MouseDown VC -
2	[B] Frees errors ² , consecutive)		+ 01 AI 1 (3)	-
E income				
aprivate: System::Void_button2_Click(System::Object^sende	, System::EventÅrgs^_e) {		(Name)	nictureBox1 Mc
if (HintP) [Access	private
Hintr = false;			CanOver	ride False
getCand():			File	c:¥texsrc¥情報器
HintP = true;			FullName	sudoku::Form1:
			IsConsta	nt False
ShowBan();			IsInjecte	d False
Contraction of Southeast (Valid a link on Paul Network Paul October 10) is an		Count Anna () (I	IsInline	True
Sprivate. Systemvord pretureboxt_wodsebown(Systemob)ect	sender, systemmindowsronmsmods	serventariss e/ (IsOverloi	ade False
			IsSealed	False
			IsShared	False
			IsTempla	ate False
			IsVirtual	False
			MustImp	lerr False
			TypeStri	ng System::Void
			~	
100 % - <			>	
出力			- 4 ×	
出力元の表示(S): デバッグ - 20 よ) ふ	2 P			
'sudoku, eve' (マネージ (v4.0.30318)): 'C:WUNDOWSWI crosoft, Net Yassenbly#GAC_MSU)	System DrawingWv4.0 4.0.0.0 b035557511d50a3aWSyste	n.Dravior.dll'が待み込まれました	0	
'sudoku.exe' (국ネージ (v4.0.30319)): 'C:WWINDOWSWNicrosoft.NetVassemblyWGAC_MSILA	System¥v4.0_4.0.0.0_b77a5c561834e089¥System.dll' #	が読み込まれました		
sudoku.exe (マネージ (v4.0.30313)): C:VVINDOVSVNICPOSOFT.NetVassenDIVVGAC_HSILA 'sudoku.exe' (マネージ (v4.0.30313)): 'C:VVINDOVSVNICPOSOFT.NetVassenDIVVGAC_HSILA	scorlib.resources¥v4.0_4.0.0.0_bustot/tildb0asa¥4ccess	iDility.dll が飲み込まれました ¥wscorlib.resources.dll が読み込まれました		
ブログラム '[4352] sudoku.exe: マネージ (v4.0.30818)' はコード 0 (0x0) で終了しま!	<i>t</i>			
< C			> 🔆 ツールボック	ス 😁 プロパティ
準備完了		380 行	115列 115文	字 挿入

}

を

```
private: System::Void pictureBox1_MouseDown(System::Object^
                                                              sender,
           System::Windows::Forms::MouseEventArgs^ e) {
    int x = e ->X;
    int y = e -> Y;
    int K = 9;
    int s = pictureBox1->Height / (K+2);
    int w = (pictureBox1->Width - s * 9) / 2;
    int h = (pictureBox1 -> Height - s * 9) / 2;
    if (x < w || x > w+9*s || y < h || y > h+9*s) {
        if (mouseDownP && mouseX >=0 && mouseY >= 0) {
            Graphics^ g = pictureBox1->CreateGraphics();
            Brush^ brush = gcnew SolidBrush(Color::White);
            g->FillRectangle(brush, w+s*mouseX, h+s*mouseY, s, s);
            Pen^ pen = gcnew Pen(Color::Black, 1);
            g->DrawRectangle(pen, w+s*mouseX, h+s*mouseY, s, s);
            mouseX = mouseY = -1;
        }
        mouseDownP = false;
        return;
    }
    int i, k;
    for (i=0; i<9; i++)
        if (x >= w+s*i && x < w+s*(i+1)) {
            break;
        }
    for (k=0; k<9; k++)
        if (y >= h+s*k && y < h+s*(k+1)) {
            break;
        }
    if (mouseDownP) {
        if (mouseX == i && mouseY == k) {
            Graphics^ g = pictureBox1->CreateGraphics();
            Brush^ brush = gcnew SolidBrush(Color::White);
            g->FillRectangle(brush, w+s*i, h+s*k, s, s);
            Pen^ pen = gcnew Pen(Color::Black, 1);
            g->DrawRectangle(pen, w+s*i, h+s*k, s, s);
            mouseDownP = false;
        } else {
            Graphics^ g = pictureBox1->CreateGraphics();
```

```
Brush^ brush = gcnew SolidBrush(Color::White);
            g->FillRectangle(brush, w+s*mouseX, h+s*mouseY, s, s);
            brush = gcnew SolidBrush(Color::Yellow);
            g->FillRectangle(brush, w+s*i, h+s*k, s, s);
            Pen^ pen = gcnew Pen(Color::Black, 1);
            g->DrawRectangle(pen, w+s*mouseX, h+s*mouseY, s, s);
            pen = gcnew Pen(Color::Black, 1);
            g->DrawRectangle(pen, w+s*i, h+s*k, s, s);
            mouseX = i;
            mouseY = k;
            mouseDownP = true;
        }
    } else {
        Graphics^ g = pictureBox1->CreateGraphics();
        Brush^ brush = gcnew SolidBrush(Color::Yellow);
        g->FillRectangle(brush, w+s*i, h+s*k, s, s);
        Pen^ pen = gcnew Pen(Color::Black, 1);
        g->DrawRectangle(pen, w+s*i, h+s*k, s, s);
        mouseX = i;
        mouseY = k;
        mouseDownP = true;
    }
}
```

```
とする。
```



Form1.h[デザイン]をクリックし、「数値入力」のボタンをダブルクリックする。



private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
 }

を

```
private: System::Void button1_Click(System::Object sender, System::EventArgs e) {
    int n = System::Convert::ToInt32(textBox1->Text);
    if (n < 0 || n > 9) {
        return;
    }
    if (mouseDownP) {
        int K = 9;
        int s = pictureBox1->Height / (K+2);
        int w = (pictureBox1->Width - s * 9) / 2;
        int h = (pictureBox1->Height - s * 9) / 2;
        Graphics^ g = pictureBox1->CreateGraphics();
        Brush^ brush = gcnew SolidBrush(Color::White);
        g->FillRectangle(brush, w+s*mouseX, h+s*mouseY, s, s);
        Pen^ pen = gcnew Pen(Color::Black, 1);
        g->DrawRectangle(pen, w+s*mouseX, h+s*mouseY, s, s);
        if (n == 0) {
            ans[mouseX][mouseY] = 0;
        } else {
            System::Drawing::Font^ drawFont = gcnew System::Drawing::Font("Arial", 24);
            Brush^ drawBrush = gcnew SolidBrush(Color::Red);
            System::String^ drawString = System::Convert::ToString(n);
            g->DrawString(drawString, drawFont, drawBrush, (float)(w+mouseX*s), (float)
                 (h+mouseY*s));
            ans[mouseX][mouseY] = n;
        }
        mouseX = -1;
        mouse Y = -1;
```

```
mouseDownP = false;
}
```

とする。



実行する。

	Form1	- 🗆 🗙
Sample		
		数值入力
		Hint

メニューの「Sample1」を選択する。

	Form1	_ 🗆 🔤 🔜 🗙 🔛
Sample		
		数值入力
	6 9	Hint
	76	
	2 8 5	
	4 8	
	3 2	
	85394	

どこかのマスをクリックする。

								F	orm1	l.				-	×	
Sample																
												۵				
		19				7	9		3			数位	植入力	I		
								7	1							
	6		0			j –	8 8		6 6			F	lint			
	_		7	6	<i>c</i>	-		~	·							
	2			8		5										
			4			8										
		3			2	÷										
			8	5	3			9	4							
	1415															

TextBox に適当な数字(例えば 5)を入力する。

	Form1	- 🗆 📉 🗙
Sample		
		5
	1 9 3	
	9 7 7	数值入力
	6 9	Hint
	3 2	
	8 5 3 9 4	

「数値入力」のボタンをクリックする。黄色のマスが5になった。

	Form1	- 🗆 🔤 📈
Sample		
		5
	1 9 3 9 7	数值入力
		Hint
	2 8 5	
	85394	

Form1.h[デザイン]をクリックし、メニューの「Sample2」をダブルクリックする。



private: System::Void sample2ToolStripMenuItem_Click(System::Object^ sender,

```
System::EventArgs^ e) {
 }
に、解きたい問題を打ち込む。
private: System::Void sample2ToolStripMenuItem_Click(System::Object^ sender,
 System::EventArgs^ e) {
    int board[9][9] = {
      \{0,0,0,9,0,8,1,0,0\},\
      \{6,2,4,0,0,1,0,0,0\},\
      \{0,0,0,0,0,0,0,0,0,0\},\
      \{0,0,8,5,0,0,0,0,9\},\
      \{0,5,0,0,2,3,0,0,0\},\
      \{0,0,7,0,0,0,3,0,1\},\
      \{9,0,0,7,0,0,8,0,0\},\
      \{0,0,0,6,0,0,9,0,3\},\
      \{5,0,0,0,0,0,6,0,0\}\};
    for (int i=0; i<9; i++) {</pre>
        for (int k=0; k<9; k++) {</pre>
             ban[i][k] = board[k][i];
             ans[i][k] = ban[i][k];
            n_cand[i][k] = n_candYoko[i][k]
                              = n_candTate[i][k] = n_candGroup[i][k] = 0;
        }
    }
    ShowBan();
```

}

🤝 sudoku - Microsoft Visual Studio Academic		- 🗆 ×
ファイル(E) 編集(E) 表示(<u>V</u>) Qt5 プロジェクト(<u>P</u>) ビルド(<u>B</u>) デバッグ(<u>Q</u>) チーム(<u>M</u>) データ(<u>A</u>) ツール(<u>T</u>) テスト(<u>S</u>) ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>)		
🔁 * 🗃 * 😂 😹 🕼 🐔 🐮 🧐 * 🖓 * 💭 👘 Debug 🔹 Win32 🔹 💋 abs 🔹 📢 🖓 🖓 🖓 🖓 🖓 🖓 🖉 🖉		
[] N. S. 水管 建建 [] P.		
離 ソリューション エクスプローラー クラス ビュー Form1.h 「デザイン」*	▼ プロパティ	+ # ×
* sandeu: Form1 - @sample2ToolStrinMenuItem Click(System::Object:Asender, System::EventArns:Ae)	sample2Tool5	tripMenuItem ,
	+ 01 AL 100	
bprivate: Śvstem::Yoid sample2ToolStripMenuItem Click(Svstem::Object^ sender, Svstem::EventArgs^ e) {	4 C++	and at a lot of
int board[9][9] = {	(Name)	samplez looistri
Y {0.0.0.9.0.8.1.0.0}.	CanOverrid	Epice
	File	c:¥texsrc¥情報器
(0.0,8,5,0,0,0,0,0),	FullName	sudoku::Form1:
[0,5,0,0,2,3,0,0,0],	IsConstant	False
[0,0,7,0,0,0,3,0,1].	IsInjected	False
[9,0,0,7,0,0,8,0,0],	IsInline	True
	IsOverloade	e False
for (int int ist ist int)	IsSealed	False
for (int k=0; k<9; k++) [IsShared	False
ban[i][k] = board[k][i];	IsTemplate	False
ansLijlkj - banLijlkj;	Isvirtual	False
n_candLiJLKJ = n_candToKoLiJLKJ = n_candlateLiJLKJ = n_candbroupLiJLKJ = 0;	Turactimpier	Faise
	TypeString	System. void
ShowBan();		
	\sim	
100 % • <		
出力	×	
出力元の表示(S): デバッグ ・ (2) (2) 😵 💌		
'sudoku, exe' (マネージ (v4.0.30319)); 'C:WINDOVSWIjcrosoft, NetWassenbl/WGAC HSILVSvsten, DravinxWv4.0 4.0.0.0 b03f577f11d50s3sKSvsten, Dravinx, d11' が読み込まれました	~	
*sudoku, exe' (マネージ (v4.0.30319)): 'C:WINDOWSMicrosoft, NetVaseeb V4060; HSTLWSystemV4.0, 4.10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,		
sudaku.exe (マネージ (v4.0.30318)): C. WINDOWSHI Crosoft. NetWassenblyWAG, Bittwiczest Dir Trywal, 4.0.0.0,Dollar, https://www.bittwiczest.org//www		
プログラム [8548] sudoku.exe: マネージ (v4.0.30318) [オコード 0 (0x0) で終了しました。	~	
	🛠 ツールボックス	🚰 プロパティ
主 编示了 484 行 32 }	20 文字	挿入

実行し、Sample2 を選択し、Hint をクリックすると

	Form1	- 🗆 ×
Sample		
		1
	³⁷ ³⁷ ³⁵ 9 ³⁴⁵ 8 1 ²³⁴ ⁶⁷⁵	
	6 2 4 ³ ³⁵⁷ 1 ⁵⁷ ³⁵⁷ ⁵⁷⁸	数值入力
	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	
	¹²³ 6 ³⁴ 8 5 ¹⁴⁶ 467 ²⁴⁷ ²⁴⁶ 9	Hint
	¹⁴ 5 ¹⁶⁹ ¹⁴⁸ 2 3 ⁴⁷ ⁴⁶⁷ ⁴⁶⁷	
	²⁴ 469 7 48 468 469 3 68 1	
	9 ¹³⁴ ¹²³ 7 ¹³⁴ ²⁴⁵ 8 ¹²⁴ ²⁴⁵	
	124 147 12 6 145 245 9 57 3	
	5 $\frac{134}{78}$ $\frac{123}{48}$ $\frac{123}{89}$ $\frac{134}{89}$ $\frac{249}{6}$ $\frac{124}{7}$ $\frac{247}{7}$	

数独の問題には著作権がありますので、確実にオリジナルと分かる非対称な問題を使っています が、普通の数独の問題で大丈夫です。

盤面を印刷でき	るようにしま	しょう。メニュー	-に「File」	を作ります。
---------	--------	----------	----------	--------

■ Sudoku - Microsoft Visual Studio アイル(F) 編集(日 表示(V) Q15 7/12/20/P) ビルド(日 デ/のグ(D) チーム(M) SQL(Q) 編式(O) アール(T) 77スト(S) 分類(M) クインド(V) ヘルプ(H) ③ - ○ (D - 4) 編 峰 (P) - マート D - フル Windows デパタガー・目動 ・ Release Wind2 ・ 通 編 - ③ の - (D - 4) 単 - ◎ (C + D - フル Windows デパタガー・目動 ・ Release Wind2 ・) 単 - ◎ (D = 1 + M + 本) (M - 2)	- 4 - 4 (Q+h
Provide Stript = ≥	
3627	

「File」の下に「Print」、「-」、「Open」、「Save」、「Save As」を作ります。

Sudoku - Microsoft Visual Studio 7rfu/F) 編集(E) 表示(V) QT5 プロ527b/(P) どルド(B) デバッグ(D) チーム(M) SQL(Q) 書式(O) ツール(T) デルド(S) 分析(N) ウィング(W) ヘルプ(H) ○・○ ③ ・ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	クイック記録(Ctrl+Q)
■ menuStrip1 筆集完了	

「ツールボックス」の「印刷」の

 ★ sudoku - Microsoft Visual Studio ブイル(F) 構築(E) 表示(V) QT5 プロジェクト(P) ○ • ○ 昭 · 留 留 ピ つ · で · ▶ ローカ ※回 ヒ ゴ 正 山 キ # 林 幸 비 工 器 	ビルド(8) デガゆグ(D) チーム(M) SQL(Q) ソール(T) デスド(5) 分析(N) クイン Windows デバッガー・目軸 ・ Release ・ Win32 ・ 厚っ ロージャード 中 中 中 中 山 ドレ 王 詩 座 中 キー 戸 伊 っ	ケイッル記載(Ctrl+Q) ク - 6* × PO(W) ヘルプ(H) 回 田 中 國 ウエルの環境の変更(Y)・ 皆 点 日 田 -
	□ · · · · · · · · · · · · · · · · · · ·	
事績完了	o	RightDuefHiz False Text Form1 UseWaitCurs False デ Text コントロールに硬連付けられたテキストで ず。

「PrintDialog」と「PrintDocument」をドラッグします。

コ コ 戸 山 キ 州 州 キ 州 m1.h [テサイン]* + ×	IN∥≡□,∜†∣⊏≑⊐⊓	- 井 正 州 王 詔 凾 井 幸 宮 踏 ≠	▼ ソリューション エクスプローラー ▼
2	Form1		
Sample File	FOUNT	1 数值入力 Hint	ソリューシュン 25/201-5-09 マリューション 25/201-5-09 マリューション 25/201400 (1 マリーション 25/20140 (1 マリーン・ション 25/2014 (1 ローン・ション 25/2014 (1 ローン・ション 25/2014 (1 ローン・ション
			(ソリニーシー・チーム エー・クラ プロ/ティー printDocument1 System 副 P→ D チール 日 その絶 DocumentNa document OriginAtMarg False

メニューの「File」の「Print」をダブルクリックします。

Sudoku - Microsoft Visual Studio	rl+Q) 👂	- • - •	₽ × ≀1; • ₽×
<pre></pre>	VIII-332 IDX VIII-332 IDX VIII-332 VIIII-332 VIIII-332 VIII-332 VIIII-33 VIIII-332 VIIII-33 VIII-33 VIII-33 VII) - さ プローラー('sudoku' 、ファイル ー ファイル orm1.h) Form1. sudoku source.h tdafx.h スファイル 依存関係 iMe.txt	■ 圖 ^(*) ○ 液 ク • (1 プロジェ! resX
	JJIIII VCCde Formi VCCde B 2+ 2 2 4 2 4 C++ (Name) Access File FullName IsAbstract C++	eClass eClass Form1 public c:¥texs sudoku False	· · · · · · · · · · · · · · · · · · ·
摩備完了 555行 3列	3 文字		挿入

```
private: System::Void printToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
```

}

に

```
if (printDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK) {
    printDocument1->PrinterSettings = printDialog1->PrinterSettings;
} else {
    return;
```

```
}
pCount = 1;
printDocument1->Print();
```

と打ち込みます。



```
int pCount;
```

```
private: System::Void printToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    if (printDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK) {
        printDocument1->PrinterSettings = printDialog1->PrinterSettings;
    } else {
        return;
    }
    pCount = 1;
    printDocument1->Print();
}
Oように、int pCount を追加します。
```

K Sudoku - Microsoft Visual Studio	ctrl+Q)
<pre>Formi_ht [#94]* formi_ht [#94]* formi_ht</pre>	
レディ 555 行 1 別	1 文字 挿入

「Form1.h「デザイン」」の下の方にある「PrintDocument1」をダブルクリックします。

	rl+Q) ・ ジリューション Iが ・ ジョン II・ ・ ジリューション II・ ・ ジョン II・ ・ ジリューション II・ ・ ジリューション II・ ・ ジョン II・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	P - なり 単 約 スプローラー で・ ぞ 首 スプローラー の ン 'sudoku' (ku スファイル ダーファイル	5 × * # × ! 圖 " 注方示:/
<pre>SoftWate: System::Noid print Document Trail CiteCoverant Copient: a server: void and trail CiteCoverant Copient: a server: void a trail citeCoverant copient coverant copient coverant copient coverant copient coverant copient coverant copient coverant coverant copient coverant copient coverant copient coverant co</pre>	▲ 画 ● ● ■ リリ ● ■ 外 ● Re イ リリューシー チ プロパティ	Porm1.n ♪ Form1.r () sudoku resource.h stdafx.h ースファイル 即依存開係 adMe.txt ーム エ… ク	esX ⇒ ∋スビュ- ▼ ₽ ×
	₩ ₩ ₩ ₩ ₩ ₩ ₩ ₩ ₩ ₩		
レディ 574 行 1 列			挿入

private: System::Void printDocument1_PrintPage(System::Object^ sender, System::Drawing::Printing::PrintPageEventArgs^ e) { } { { if (pCount > 0) { Graphics^ g = e->Graphics; int K = 9;

int s1 = e->PageBounds.Width / (K+2);

```
int s2 = e->PageBounds.Height / (K+2);
int s = (s1 > s2) ? s2 : s1;
int w = (e->PageBounds.Width - s * 9) / 2;
int h = (e->PageBounds.Height - s * 9) / 2;
Pen^ pen = gcnew Pen(Color::Black, 1);
for (int i=0; i<=9; i++) {</pre>
   if (i % 3 == 0) {
      pen = gcnew Pen(Color::Black, 3);
      g->DrawLine(pen, w, h+i*s, w+9*s, h+i*s);
   } else {
      pen = gcnew Pen(Color::Black, 1);
      g->DrawLine(pen, w, h+i*s, w+9*s, h+i*s);
   }
}
for (int i=0; i<=9; i++) {</pre>
   if (i % 3 == 0) {
      pen = gcnew Pen(Color::Black, 3);
      g->DrawLine(pen, w+i*s, h, w+i*s, h+9*s);
   } else {
      pen = gcnew Pen(Color::Black, 1);
      g->DrawLine(pen, w+i*s, h, w+i*s, h+9*s);
   }
}
for (int i=0; i<9; i++) {</pre>
   for (int k=0; k<9; k++) {</pre>
      if (ban[i][k] > 0) {
         System::Drawing::Font^ drawFont = gcnew System::Drawing::Font("Arial", 45);
         Brush^ drawBrush = gcnew SolidBrush(Color::Black);
         System::String^ drawString = System::Convert::ToString(ban[i][k]);
         g->DrawString(drawString, drawFont, drawBrush, (float)(w+i*s), (float)(h+k*s));
      } else if (ans[i][k] > 0) {
         System::Drawing::Font^ drawFont = gcnew System::Drawing::Font("Arial", 45);
         Brush^ drawBrush = gcnew SolidBrush(Color::Red);
         System::String^ drawString = System::Convert::ToString(ans[i][k]);
         g->DrawString(drawString, drawFont, drawBrush, (float)(w+i*s), (float)(h+k*s));
      }
   }
}
if (HintP) {
   for (int i=0; i<9; i++) {</pre>
      for (int k=0; k<9; k++) {</pre>
         if (ans[i][k] == 0) {
            System::Drawing::Font^ drawFont = gcnew System::Drawing::Font("Arial", 15);
```

```
Brush^ drawBrush = gcnew SolidBrush(Color::Blue);
               for (int j=0; j<n_cand[i][k]; j++) {</pre>
                  System::String^ drawString = System::Convert::ToString(cand[i][k][j]);
                  if (j < 3)
                     g->DrawString(drawString, drawFont, drawBrush,
                              (float)(w+(i+0.2*j)*s), (float)(h+k*s));
                  else if (j < 6)
                     g->DrawString(drawString, drawFont, drawBrush,
                               (float)(w+(i+0.2*(j-3))*s), (float)(h+(k+0.3)*s));
                  else
                     g->DrawString(drawString, drawFont, drawBrush,
                               (float)(w+(i+0.2*(j-6))*s), (float)(h+(k+0.6)*s));
               }
            }
         }
      }
   }
}
```

を打ち込みます。これで、盤面が印刷できますが、私のパソコンでは OS が 64 ビットなので、 Win32 ではプリンタの選択が出来ず、Release x64 でコンパイルして、使っています。

次に、盤面を保存したり、保存したものを読み込んだりできるようにしましょう。「ツールボッ クス」の「ダイアログ」の

Sudoku - Microsoft Visual Studio Zr4/k(F) 編集(E) 表示(V) QT5 プロシェクト(P) G・0 間・留 目 ピーク・ペート ロース 版(上 コ 正 止) キ # 株 主 料 工 語	クイタウル ビルド(B) デブ(G*)(D) チーム(M) SQL(Q) ダール(T) デスト(S) 分析(N) タインドグ(M) ヘルプ(H) カル Windows デパップー・目動・・Release・X64・・ 月 _ま ② 四 田 中 酉 ケエンの種類の変更(Y)・皆 点 [回 二 素 卡 ロ キ コ ロ キ 山 M 工 読 香 井 キ ダ ダ 。	動 (Ctrl+Q) 오 - 6 X
Control Control Contro Contro Control Control Control Control	Form1	
李備 完了	rintDocument1	Image PictureBox に表示されるイメージで す。

「SaveFileDialog」と「OpenFileDialog」をドラッグします。

Joku.cpp Form1.h*	Form1.h [デザイン]* キ ×		▼ ソリューション エクスプローラー ▼
	Form1		
Sample File		1 数值入力 Hint	マリユーション Sudoku (1 ガ ・ 留 シリスフイル ・ 留 シッスフイル ・ 留 シッスフィル ・ 目 シッスフィル ・ 目 Sudoku ・ B resourceh ・ B stadskh ・ 動 stadskh ・ 観 ジッスフィル ・ 聞 Swafe現紙 ■ ReadMe.bt
			⊞ (ApplicationS

メニューの「Save」と「Save As」をプログラミングします。メニューの「Save」と「Save As」 をそれぞれダブルクリックします。

く sudoku - Microsoft Visual Studio 24/2018 フィッパロー 現在(日、東市内) ロマ Thistophynia Pellatian エーバル Spirian マーパロ マーパロ マンドロ AMERIA のへおのいれ ANT/U	th (Ctrl+Q) 👂 – 🗗 🗙
アイカロ ● ● ● ● ● ● ● ● ● ● ● - ○ - ▶ ● - かみ Windows デバック - 自動 - Release - 水谷 - ● ● ● ● ● ● ● ● ● ● ブッの準約の表現() - '客 会 [: 全句 = ○ ● ● ● ● ● ● ● ● ● - つかみ Windows デバック - 自動 - Release - 水谷 - ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	:a, b a \$ \$ ∎ 111‡
	▼ ソリューション エクスプローラー ▼ 単 ×
👎 🔩 sudoku::Form1 🔹 🗣 saveAs ToolStripMenuItem_Click(System::Object ^ sender, System::EventArgs ^ e)	· 000 0-200
10	↓ ソリューション エクスプローラー の検 ・
J J J	
100 % • 4	saverooistripmenuitem_clic *
出力 出力の(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	
レディ 638行 10	02列 102文字 挿入

private: System::Void saveToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {

```
}
```

に

if (!fname) {
 saveAsToolStripMenuItem_Click(sender, e);
 return;

```
}
StreamWriter^ swriter = gcnew StreamWriter(fname);
System::String^ str = "";
for (int k=0; k<9; k++) {</pre>
   str = "";
   for (int i=0; i<9; i++) {</pre>
      str += ban[i][k].ToString();
      if (i < 8)
         str += " ";
   }
   swriter->WriteLine(str);
}
for (int k=0; k<9; k++) {</pre>
   str = "";
   for (int i=0; i<9; i++) {</pre>
      str += ans[i][k].ToString();
      if (i < 8)
         str += " ";
   }
   swriter->WriteLine(str);
}
swriter->Close();
```

```
と打ち込み、
```



private: System::Void saveAsToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {

}

に

```
SaveFileDialog^ sfdlg = gcnew SaveFileDialog();
if (sfdlg->ShowDialog() != Windows::Forms::DialogResult::OK) return;
fname = sfdlg->FileName;
StreamWriter = gcnew StreamWriter(fname);
System::String^ str = "";
for (int k=0; k<9; k++) {</pre>
   str = "";
   for (int i=0; i<9; i++) {</pre>
      str += ban[i][k].ToString();
      if (i < 8)
         str += " ";
   }
   swriter->WriteLine(str);
}
for (int k=0; k<9; k++) {</pre>
   str = "":
   for (int i=0; i<9; i++) {</pre>
      str += ans[i][k].ToString();
      if (i < 8)
         str += " ";
   }
   swriter->WriteLine(str);
}
swriter->Close();
```

```
と打ち込みます。
```



そして

using namespace System::IO; using namespace std;

をプログラムの上部に打ち込みます。



さらに、

System::String^ fname;

を今打ち込んだプログラムの上に書き込みます。



これで盤面が保存できるようになりました。

								F	orm1	L			-	×
Sample	File													
										3		4		
	457 8	1	256	24	456 8	246	9	245 68	3			-		
	345 8	9	235 6	123 4	145 68	7	245 68	245 68	256 8			数值入力	,	
	345 8	256 8	235 6	234 9	456 89	234 69	245 68	7	1					
	6	58	9	123 47	147	123 4	123 457	123 458	257 8			Hint		
	135 8	58	7	6	149	123 49	8 123 458	123 458	258 9			L		
	2	4	13	8	179	5	136 7	136	679					
	157	256	Δ	179	167 9	8	123 567	123 56	256 7					
	157	2	156	147 9	2	146	156 78	156	567					
	17	267	0	5	2	16	126	0	1					
			0	J	J		(3	4					

の盤面を「Save As」で保存すると

0	1	0	0	0	0	9	0	3
0	9	0	0	0	7	0	0	0
0	0	0	0	0	0	0	7	1
6	0	9	0	0	0	0	0	0
0	0	7	6	0	0	0	0	0
2	0	0	8	0	5	0	0	0
0	0	4	0	0	8	0	0	0
0	3	0	0	2	0	0	0	0
0	0	8	5	3	0	0	9	4
0	1	0	0	0	0	9	0	3
0	9	0	0	0	7	0	0	0
0	0	0	0	0	0	0	7	1
6	0	9	0	0	0	0	0	0
0	0	7	6	0	0	0	0	0
2	4	0	8	0	5	0	0	0
0	0	4	0	0	8	0	0	0
0	3	0	0	2	0	0	0	0
0	0	8	5	3	0	0	9	4

というデータが書き込まれています。単に *ban*[][] と *ans*[][] の数字が並んでいるだけです。これを 読み込んで

								F	orm:		×
Sample	File										
	2			_			_		-	4	
	457 8	1	256	24	456 8	246	9	245 68	3		
	345 8	9	235 6	123 4	145 68	7	245 68	245 68	256 8	数值入力	
	345 8	256 8	235 6	234 9	456 89	234 69	245 68	7	1		
	6	58	9	123 47	147	123 4	123 457	123 458	257 8	Hint	
	135 8	58	7	6	149	123 49	8 123 458	123 458	258 9		
	2	4	13	8	179	5	136 7	136	679		
	157 9	256 7	4	179	167 9	8	123 567	123 56	256 7		
	157 9	3	156	147 9	2	146 9	156 78	156 8	567 8		
	17	267	8	5	3	16	126 7	9	4		
	1										

の盤面を復元できるようにします。 メニューの「Open」をダブルクリックします。

Yuz-ball Yuz-ball Yuz-	Y sudoku - Microsoft Visual Studio ケイック活動(ファイル(E) 構築(E) 表示(Y) のです。プロシェカト(E) どんド(E) デパッグ(D) チーム(M) SQL(D) ツール(I) テスト(S) 分析(M) ウィンドク(M) へんげ(H) マーレーローローローローローローローローローローローローローローローローローロー	ctrl+Q)
100% · · · · · · · · · · · · · · · · · ·	<pre></pre>	・ ○ ① ③ ● ④ ③ ● ● ● ③ ・ ジリューション エウスブローラー の後 ● ・ ・ ジリューション ンロカスパローラー の後 ● ・ ・ ジョン ンロカスパール ・ ③ ・ コムロカル ・ ③ ・ コムロカル ・ ③ ・ コムロカル ・ ③ ・ コムロカル ・ ⑤ ● ⑤ ・ コース フイパル ・ ③ 「の 「 「 「 「 」 」 ・ ⑤ ● ⑤ ・ コース フイパル ・ ⑤ ● ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
	100% - * * * * * * * * * * * * * * * * * *	x = 24 C++ (Name) openToolStri, Access private CanOverride False File c.*texsrc¥fil FullName sudoku::Forr* C++

```
private: System::Void openToolStripMenuItem_Click(System::Object^ sender,
   System::EventArgs^ e) {
 }
に
OpenFileDialog^ ofdlg = gcnew OpenFileDialog();
if (ofdlg->ShowDialog() != Windows::Forms::DialogResult::OK) return;
fname = ofdlg->FileName;
StreamReader^ sreader = gcnew StreamReader(fname);
String^ delimStr = " ,";
array<Char>^delimiter = delimStr->ToCharArray();
for (int k=0; k<9; k++) {</pre>
   array<String^>^ splitString = sreader->ReadLine()->Split(delimiter, 60);
   for (int i=0; i<9; i++) {</pre>
      ban[i][k] = int::Parse(splitString[i]);
   }
}
for (int k=0; k<9; k++) {</pre>
   array<String^> splitString = sreader->ReadLine()->Split(delimiter, 60);
   for (int i=0; i<9; i++) {</pre>
      ans[i][k] = int::Parse(splitString[i]);
   }
}
sreader->Close();
for (int i=0; i<9; i++)</pre>
   for (int k=0; k<9; k++) {</pre>
      n_candTate[i][k] = n_candYoko[i][k] = n_candGroup[i][k] = n_cand[i][k] = 0;
```

```
}
mouseDownP = false;
mouseX = 0;
mouseY = 0;
HintP = false;
ShowBan();
```

と打ち込みます。



先ほど保存した盤面を読み込んでみます。

•								F	orm1	Ľ	 			-	×	2
Sample	File															
												1				
		1	6 	6	49 10		9	ic C	3							
		9		_		7						数值	入力			
				5	2	2		7	1							
	6		9	6		4i T		ю				Hi	int			
		•	7	6	·	6		e	e							
	2	4		8		5	-									
	6		4	e e		8	6 0	а С	0							
		3		e	2	6	e	·								
			8	5	3	2	g.	9	4	5						

デバッグする時は、例えば

```
System::String^ str = "";
str = "i=";
str += i;
str += " k=";
str += k;
MessageBox::Show(str);
というコードを
private: System::Void pictureBox1_MouseDown(System::Object^ sender,
   System::Windows::Forms::MouseEventArgs^ e) {
   int x = e -> X;
   int y = e -> Y;
   int K = 9;
   int s = pictureBox1->Height / (K+2);
   int w = (pictureBox1 -> Width - s * 9) / 2;
   int h = (pictureBox1 -> Height - s * 9) / 2;
   if (x < w || x > w+9*s || y < h || y > h+9*s) {
      if (mouseDownP && mouseX >=0 && mouseY >= 0) {
         Graphics^ g = pictureBox1->CreateGraphics();
         Brush^ brush = gcnew SolidBrush(Color::White);
         g->FillRectangle(brush, w+s*mouseX, h+s*mouseY, s, s);
         Pen^ pen = gcnew Pen(Color::Black, 1);
         g->DrawRectangle(pen, w+s*mouseX, h+s*mouseY, s, s);
         mouseX = mouseY = -1;
      }
      mouseDownP = false;
      return;
   }
   int i, k;
   for (i=0; i<9; i++)
      if (x >= w+s*i && x < w+s*(i+1)) {
         break;
      }
   for (k=0; k<9; k++)
      if (y >= h+s*k && y < h+s*(k+1)) {
         break;
      }
System::String^ str = "";
str = "i=";
str += i;
str += " k=";
```

```
str += k;
MessageBox::Show(str);
   if (mouseDownP) {
      if (mouseX == i && mouseY == k) {
         Graphics^ g = pictureBox1->CreateGraphics();
         Brush^ brush = gcnew SolidBrush(Color::White);
         g->FillRectangle(brush, w+s*i, h+s*k, s, s);
         Pen^ pen = gcnew Pen(Color::Black, 1);
         g->DrawRectangle(pen, w+s*i, h+s*k, s, s);
         mouseDownP = false;
      } else {
          Graphics^ g = pictureBox1->CreateGraphics();
          Brush^ brush = gcnew SolidBrush(Color::White);
          g->FillRectangle(brush, w+s*mouseX, h+s*mouseY, s, s);
          brush = gcnew SolidBrush(Color::Yellow);
          g->FillRectangle(brush, w+s*i, h+s*k, s, s);
          Pen^ pen = gcnew Pen(Color::Black, 1);
          g->DrawRectangle(pen, w+s*mouseX, h+s*mouseY, s, s);
          pen = gcnew Pen(Color::Black, 1);
          g->DrawRectangle(pen, w+s*i, h+s*k, s, s);
          mouseX = i;
          mouseY = k;
          mouseDownP = true;
      }
   } else {
      Graphics^ g = pictureBox1->CreateGraphics();
      Brush^ brush = gcnew SolidBrush(Color::Yellow);
      g->FillRectangle(brush, w+s*i, h+s*k, s, s);
      Pen^ pen = gcnew Pen(Color::Black, 1);
      g->DrawRectangle(pen, w+s*i, h+s*k, s, s);
      mouseX = i;
      mouseY = k;
      mouseDownP = true;
   }
}
の様に打ち込み、マウスをクリックすると
```



のように、プログラムを中断して、知りたい変数の値を表示させることが出来ます。

プログラムの説明はしませんが、気になるなら自分で調べて下さい。自分で見付けたことは宝物 になります。これだけ知っていれば、これを雛形として、「ひとりにしてくれ」、「ぬりかべ」、「ま しゅ」、「カックロ」、「フィルオミノ」、「四角に切れ」「スリザーリンク」、「美術館」、「へやわけ」、 「のりのり」などのプログラミングもすることが出来ます。必要な VC++特有の事柄は説明しまし たから、後は、通常の C++とアルゴリズム(特に昔、人工知能と呼ばれていた探索のアルゴリズ ムを中心に)とデータ構造とパズルそのもの(人間のエキスパートはどのようにして論理的に解を 見つけているか、どのようにしてパズルの問題を作っているか)の勉強すればいいです。

将棋や囲碁のプログラミングと違って、パズルのプログラミングは「機械学習」以前のアルゴリ ズムだけを使って解くことができます。これらのアルゴリズムはもはや特別なものではなく、枯れ た技術として常識になっているので、パズルのプログラミングは誰でも出来ます。数独の問題を作 るプログラミングは簡単ですので、「四角に切れ」の問題を作成するプログラムはどのようにプロ グラミングすればよいか、私がやったことを解説します。

作った問題がユニークな解をもつかどうか判定できるように、まず「四角に切れ」の問題をバッ クトラッキングで解がないか、解が一つか、二つ以上か判定できる関数を作ります。人間のエキス パートはどのようにして論理的に解を見つけているかの知識を使って簡単な問題なら解けるように プログラミングしておき、バックトラッキングの実行時間を短くするために、その関数を使って枝 刈りします。準備が出来たら、「四角に切れ」の問題作成のプログラミングを始めます。

まず私が考えた方法は、まず矩形を敷き詰め、数字をセットする方法です。まず、

HIKAKU:Form1	 [©]_e button 16_Oick(System::Object ^ sender, System::EventAn 	gs ^ e)	- 00 0 10 - ≠ 0 10 0 + ≠ -
<pre>/ ref (H = 1 % (- value; (rr) { ref (H =</pre>	20 20 4 4 4 8 8 8 10 10 10 10 8 8 8 10 10 10 10 3 8 8 8 10 10 10 10 3 8 8 8 10 10 10 10 3 8 8 8 6 6 10 10 10 3 8 8 8 6 6 6 10 10 2 6 6 6 6 6 10 10 2 6 6 6 6 10 10 2 6 6 6 6 6 10 10 2 6 6 6 6 10 10 2 2 6 6 6 10 10 10 2 2 6 6 6 6 6 10 10 10 10 10 10 10 10 10 10 10 10<	BEDSAUGOV	 yys-y-y 2375-4 dest (Are) yys-y-y seruc(17) (70) y set-3270 y set-3270

のような矩形の敷き詰めを虱潰しの方法で探索で求め、次に各矩形の数字を一個にします。解がユ ニークにするために、どの数字を残すかも虱潰しの方法で探索します。このとき、全ての組み合わ せを確かめるのでは時間がかかり過ぎるので、枝刈りします。具体的に言うと



に於いて、左上隅のサイズ8の二個の矩形や左下のサイズ2の二個の矩形や中央下のほうのサイズ 3とサイズ6の矩形の組み合わせなどに於いて、数字の配置によっては、二通りの矩形の配置が可 能です。サイズ3の3個の矩形で作られる正方形でも同じことが起こります。二通りの矩形の配置 が可能な数字が現れれば、、枝刈りします。同じ、矩形の配置でも



のように、複数の問題が作れます。この方法で問題は作成できたが、ニコリの問題集は



のような数字の配置が点対称な問題が載っています。これは後で説明する方法でコンピュータが 作った問題です。このような問題を作る方法を考えてみます。

不幸なことにこの為に、私が最初に考えた方法はランダムに点対称に数字と対応する矩形を配置 していく方法です。

(デザイン) Form1.h + × IKU:Form1	 [0], getFinalPosition2() 	 ソリューション エクスプローラー 〇 〇 〇 〇 〇 〇 - 年 〇 18 〇 0 月
<pre>ideal cafication() { the cafication() {</pre>	Image: Sector with the constrained of t	Yiyy-zy 2027/17->- 048 € Con-1 Yiyy-zy 2020/17-2727) S \$100,00 Y
ם - אול אול 1	マジェレルスコープルークポイント、コマンドウクリアク、(1971年1-5 ペットウ、出力)	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

このとき、

if (not)[10] a. M. And (100) if (not)[10] a. M. And (100) <	プロセス [5140] SHIKAKULe Form1.h (デザイン) For な SHIKAKUsForm1	ee v (1 + DF v 2k)>F mlh + × candidates[ind2] = candidates[n_candidates - 1 , n_candidates:	- て ボ 第2,2922/2-2	- a x]	ジリコーション エクスプローラー ・ キ × ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
	000 % •]	<pre>f Gran(1)D) == 0 & A moleck = 1 - 1[1ats =</pre>	is Series 5 4 4 5 9 4 4 5 9 4 4 5 9 4 4 5 9 4 4 5 9 6 6 9 1010101010 6 6 9 1010101010 6 2 9 8 8 8 6 9 3 8 8 6 9 6 6 6 6	B B(tr): Mile B B(tr):	 ○ 1/2 - 2/2 SPACUAT (17/25/27) ○ 1/25/7/6 ○ 1/25/7/6

のように、サイズ1の正方形が残る場合は、あと続けても無駄であるから枝切りする。更に、

ロ(デザイン) Fo GAKUt:Form1	mih + ×	[©] _e getFinalPosition2()	
• <	<pre>cadd state [[mt]] 2 = candidate in_randidates f.add state [[mt]] 2 = candidates in_randidates in_randidates f.add state [[mt]] 2 = candidates in_randidates in_randidates f.add state [[mt]] 2 = candidates in_randidates in_ra</pre>	State 1000000000000000000000000000000000000	

の左辺のサイズ3のスペースの矩形ように、点対称な位置にスペースがなければ、あと続けても無 駄であるから枝切りする。更に、

Formith (F#42) Formith #	* 101 *# * AV2P	1 L W W222 N.T	 >)///3-7/21/10-5- >)//
emila (973) Femila e § SIGAUJ-femil	$ \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\$	- 0 gerfindhouted) # 24xxx1x sector 16 by Clanck/LALAMSA 56x6 biosectly 2017/26/26/47 TR 0 rande Exercise 5 4 4 2 4 4 4 4 5 9 4 4 2 4 4 4 4 5 9 10 10 6 6 6 6 5 9 10 10 7 5 9 10 10 6 6 6 6 5 9 10 10 7 5 9 10 10 6 6 6 6 5 9 10 10 7 5 9 10 10 6 6 6 6 5 9 10 10 7 5	(b) =
20 % • 4	and (11) a set of a	9 1010101010100006 9 1010101010106 9 9 8 8 6 9 10101010106 9 8 8 8 6 9 10101010106 9 8 8 8 6 9 10101010106 9 8 8 8 6 9 101010100 101010000 1010100000 10101000000 10101000000 101010000000 101010000000 101010000000000 101010000000000000000 101010000000000000000000000000	907) Nora 41 - 2 21
1955年 1955年 ローカル ウオッチ1 1957 1957	а #7(*#4), Ф. Р.		12 12 10 10 10 10 10 10 10 10 10 10 10 10 10

のように、サイズ2以上の矩形を敷き詰めることの出来ないスペースの領域が出来たら、あと続け ても無駄であるから枝切りすべきであるが左下のような不規則な形は判定が厄介である。右上のよ うな形だけチェックする。

しかし、このような小手先の単純な枝刈りでは、計算時間の短縮は望めなかった。

20 SHLARU(東行寺) - Microsoft Visual Studio 77 (LM) 編集時 表表的 720 22 MP 込代物 折けの方向 チームの アール(の 0 - 0) (○ - 1) 単形(○ - 0) - 10 形(○ - 0) - 10 日前 - 10 月前 - 11 月 70 日本 (1655) SHKARLee - 10 日本 - 20 月前 - 20 月前	FALS 分前的 90790W A&700 ● ○ 1回 → ← ← ⊂ c1 客(1回 回 国 国 国 国 国 国 国 国 国 国 国 国 国 国 国 国 国 国		□ ▼1 2/92版(Cort+Q) P = 8 × ▲ 中将治・
7 70 E2 (0040 09700.4000 7 50 000.5000 7 50 000.5000 7 50 000.5000 7 50 000.5000 7 50 00000 7 50 00000 7 50 00000 7 50 000000 7 50 000000 7 50 000000 7 50 0000000 7 50 0000000 7 50 00000000 7 50 000000000 7 50 000000000 7 50 000000000 7 50 000000000 7 50 0000000000000000 7 50 0000000000000000000000 7 50 0000000000000000000000000 7 50 0000000000000000000000000000000000	Image: Second state Image: Second state Image: Second state Image: Second state <th></th> <th>1979-50-27270-5 1970-50-988-000 (770/25) 1970-50-988-000 1970-50-50-50-50-50-50-50-50-50-50-50-50-50</th>		1979-50-27270-5 1970-50-988-000 (770/25) 1970-50-988-000 1970-50-50-50-50-50-50-50-50-50-50-50-50-50
(1) 전 전 전 전 전 전 전 전 전 전 전 전 전 전 전 전 전 전 전		رة (الر	ソリューション 10スプローテー イベル 12スプローラー イベル 12スプローラー 4005 行 44 男 5 文字 挿入 1646 ヘー 40 見一 厚 (2) 2016(17)

のような画面が延々と続きます。

これを回避するためには、これらを一般化し、面倒でもスペースの領域をサイズ2以上の矩形で 分割し、すべての分割を求める。分割が不可能であれば、枝刈りする。可能であれば、どれかの分 割に対して、それぞれの矩形のどれかの点の対称な位置にスペースがあるかどうか調べる。もし どの分割に対してもこのことが不可能なら枝刈りする。但し、このチェックは時間がかかるので、 スペースの領域のサイズが大きいときは、役立ちそうな特徴的な特徴をチェックし、枝刈りしま す。このように人間がすぐ気付くことはそれがどんなに厄介でもプログラミングしなければ、コン ピュータを使っていても計算時間の短縮は望めません。



のような問題が作れるようになりました。しかし、大きい盤面では問題が作れません。「矩形によ る分割が可能であれば、どれかの分割に対して、それぞれの矩形のどれかの点の対称な位置にス ペースがあるかどうか調べる。」では不十分です。「矩形による分割が可能であれば、どれかの分 割に対して、それぞれの矩形のどれかの点の対称な位置にスペースがあり、そのスペースを含むサ イズ2以上の矩形がそれぞれ重ならずに存在するかどうか調べる。」必要があります。このような 分割が一つあれば良いので、すべての分割の集合を求める必要はありません。プログラムを修正し ます。

更に、



のような局面は解が複数あります。これも無駄であるから枝切りする。更に、サイズ3の矩形が3 個並んで出来る正方形の配置の場合も枝刈りすべき局面があります。8×8のような小さな盤なら



のような問題を11秒ぐらいで作ってくれますが、10×10の盤では数時間掛かっても問題は作っ てくれません。虱潰しするとき、数字の位置と矩形と二つのパラメータが変化するので、盤が大き くなると探索空間が膨大になり、少々枝刈りしたぐらいでは、パソコンでは探索しきれません。ニ コリの問題を調べて直してみると単に点対称なだけではなく、随分綺麗な幾何学的配置の問題がた くさんあります。手で問題を作っている人たちは、まず、数字を配置すべき位置をまず配置し、こ の位置は後は動かさず、この位置を含む矩形達を虱潰しに配置して、問題を作っています。そこ で遅ればせながら、ランダムに点対称な数字を配置すべき位置を配置し、この位置を含む矩形達を 虱潰しに配置して、問題を作るプログラムを作ります。

才能のある人は最初からこのようなことは気付くと思いますが、このように回り道しても、時間 を短縮するために、色々試行錯誤して、色々なプログラミングをしたので、得ることはあります。 多分。人はあらゆる失敗を経験するとコンピュータが使えるようになるそうです、ある意味、予備 のプログラミングの練習をしたので、すでに作っている関数を流用できるので、このプログラミン グは簡単です。



のような問題を44秒ぐらいで作ってくれました。いちいち使っている枝刈りの技法を説明しませ んが、計算機がやっている過程を表示して、眺めていれば、人間はこのような場合、枝刈りしてい るだろうと思うことで、プログラミングが楽で、計算時間が早くなりそうなことをプログラミン グします。一般化して苦労してプログラミングしても、枝刈りすべきかどうかチェックするのに時 間がかかり、全体として、時間の短縮効果がない場合が結構あります。基本的に、数学は完璧でな ければいけませし、論理矛盾がなければどんなものでも良い(実際は美しいものが好まれますが) ですが、プログラミングはバグがあってはいけませんが、数学的に完璧である必要はありません。 数学的に完璧に作る(完全に無駄なく枝刈りする)より、バグが無いとの前提で、兎も角早く計算 してくれるプログラムを作ることが重要です。数学者である私にはこのさじ加減が難しくて、細部 に拘り、いつも無駄な努力を繰り返しています。

数字を配置すべき位置をランダムに綺麗な幾何学的に配置するよう計算機に命令するには、現在 の私の能力を超えているように思うので、人間が数字を配置すべき位置を配置した後の探索を計算 機がするプログラムを作ります。綺麗な配置は、大域的にも、局所的にも対称性の豊富な配置です。



のような問題を4分18秒ぐらいで作ってくれました。時間は単なる目安です。乱数を使って虱潰 ししているので運が悪ければすごく時間がかかります。多分、一様、完成です。これは「ペンシル パズル本32 四角に切れ1」19 作:真良碁 と同じモチーフの問題ですが、勿論数字は違って います。一つのモチーフから複数の問題が作れます。このモチーフはローズ群 D4 の対称性を持っ ています。モチーフの美しさと問題の難易度は関係ありません。

所で、「四角に切れ」と同じように思える「フィルオミノ」ですが、解析や問題作成のプログラ ミングをしてみると、公表されている問題を解くプログラムは何とか作れますが、問題作成は難 しいです。「フィルオミノ」の問題集が4冊出版されていますが、問題を作った人達がどのように して「短時間で解ける問題」を作成できたかパズル制作を始めたばかりの私には謎のままです。イ ンターネットで調べる(八登祟之著:「NP 完全なペンシルパズルの一覧」)と「フィルオミノ」は 「NP-完全」で「ASP-完全」と書いてありました。「ASP-完全」であれば、

● パズル製作者が、自作の問題とその解の1つを持っているとする。このとき、別解があるか を判定するのは NP-完全

● 解が幾つあるか判らない問題に対して、解があるかを判定するのは NP-完全

● 唯一解をもつことが判っている問題に対して、その解を求めるのは、やはり NP-完全と同 様に難しい

● 与えられた問題が唯一解(複数解や解なしでなく)であるかを判定するのは NP-完全より 少し難しい

● 与えられた問題が解を幾つもつかを求めるのは NP-完全よりずっと難しい だそうです。

2018年3月、古希になって、「情報数学」や「確率論」の講義から解放されたので、再び、「フィ ルオミノ」の問題作りに取り組みました。どんなプログラムを作っていたか完全に忘れていたの で、読み返してみると、「フィルオミノ」を解くプログラムに何か所か致命的なバグ(酷かったの は、「状況によっては無限ループになる」や「サイズ N 以下の可能なブロックの集合を集めると き、Queue に同じブロックか(順序が違うだけか)のチェックしていなかったので、時間が掛かり 過ぎていただけでなく、N=10 以上の時、メモリーを使いきっていた」など)があって、単純な方 法による問題作成やバックトラッキングにも時々失敗していました。これらを修正することによ り、バックトラッキングの上手い枝刈りの手段は思いつきませんが、フィルオミノの人間が解く方 法を大量にプログラミングしているので、バックトラッキングなしでニコリの10×18の問題が殆 ど解けますし、それを使って、バックトラッキングを高速化できました。「四角に切れ」の最初に 述べたように、解を作って、数字を点対称に極大まで消す方法では



のような問題が26分ぐらいで作ってくれるようになりました。また、



のような問題を 10 分ぐらいで作ってくれますが、表示されている数字の数が多すぎて、ニコリの パズル集に載っているようなきれいな問題にするのは偶然に頼っているので、制御するのは難しい です。



を元に、「四角に切れ」の真似をして、ブロックを虱潰しで割り当ててみると



13 秒で作ってくれます。



を元にすると


を17分18秒秒で作ってくれます。



を元にすると



を3分29秒秒で作ってくれます。



を元にすると



を3時間41分55秒秒で作ってくれます。これが唯一解

llomino ve	r. 1.0 by C	samu Nakamu	ıra (Kochi Un	iversity) 201	3/5/19						
Sample	Rei	Example P	roblem le	stData					 ・数値入力 	✓ 探索範囲の刺服 縦 7 横 7	
	5	5	5	2	6	3	3		5 数值入力	^{数字割合} 0.45 最大の数字 6	
	6	5	1	2	6	5	3		Reset	SpecialD	Backtracking(Pri Two)
	6	5	6	6	6	5	1	1	First Step Simple	SpecialD Only SpecialE	数字の点の配置
	6	1	5	5	6	5	5		Simple Only	SpecialE Only	数字の点の配置か(問題作成
	6	6	3	5	5	6	5		SpecialA Only	SpecialF SpecialF Only	数字の点の配置が問題作成く改良版
	6	1	3	3	5	6	6		SpecialB SpecialB Only	ゴロックにおいたか	間間に成
	4	4	4	4	6	6	6		SpecialC	ブロックと数字の配置	Backtracking (Mo Than Two)
				1		1		,	SpecialC Only	=1:2分 4秒515	Backtraking(One Solution) Backtracking(Al Solutions)

を持つことを調べるのに私のパソコンで、私の作ったプログラムでは2分4秒かかります。時間が 掛かり過ぎています。原因は、私の力では、論理的に考えて



ここまでしか解けないことにあります。殆ど解けないので、虱潰しの探索になり、ルールが単純な ので、枝刈りの有効な方法を思いつかないです。乱数を使っているので、運が良ければすぐに作っ てくれますが、運が悪ければ、数時間かかります。制御するのは難しいです。矩形に比べ、ブロッ クは数が多く、探索空間が膨大になって、これでは 10 × 10 の盤の問題など夢のまた夢です。

また、パズルのエキスパートが「フィルオミノ」の問題を作っている1つの方法は、多分次のよ うな方法です。



を元に問題を作ってみる。



のように作っていく。



と続けていく。左上隅は図のように522とは出来ない(2と5とが交換可能になるから、このようにしたければ、インデックス1の5の端点を2でプロテクトする必要がある)。例えば、



と続ける。



と続ける。最後に

mple	Example	Problem	TestData							
								 新値入力 や入力 	★ 株素単田田の制成 縦 10 横 10	
1	2	2	1	3	3	3	1	○ 基数入力 5	数字割合 日本の数字	
	5	5	5	5	1	5	5	数恤人力 Reset	SpecialD	Backtra
	5	3	3	3	5	5	5	First Step	SpecialD Only	数字の
	4	4	4	4	3	3	3	Simple Only	SpecialE SpecialE Only	数字の点 問題
	5	5	5	6	6	1	6	SpecialA SpecialA Only	SpecialF	問題作
	5	4	2	2	6	6	6	SpecialB	SpecialF Only	問題作成
	5	4	- 1	4	2	2	1	SpecialB Only SpecialC	ブロックと数字の 配置	Backtra Tha
	0	-	-	-	2	2	<u>'</u>	SpecialC Only	ブロックと数字の配置から問題作成	Backtr So
								number of soluti	on =2:0分 0秒522	Backtr

と問題を作る。しかし、Backtracking で調べると

Fillomino ve e Sample	r. 1.0 by Osan Example	nu Nakamur Problem	a (Kochi Uni TestData	versity) 2013	/5/19						- 0
									 ● 数値入力 ○ 枠入力 ○ 恭敬入力 	 「探索範囲の制限 縦 椎 10 	
	2	2	1	3	3	3	5		5 数值入力	^{数字割合} 0.45 最大の数字 8	
	5	5	5	5	2	2	5		Reset	SpecialD	Backtracking(F Two)
	5	3	3	3	5	5	5		First Step Simple	SpecialD Only	数字の点の配置
	4	4	4	4	3	3	3		Simple Only	SpecialE Only	数字の点の配置 問題作成
	5	5	5	6	6	1	6		SpecialA SpecialA Only	SpecialF	問題作成II(成 称)
	5	4	2	2	6	6	6		SpecialB		問題作成(点対 縦は偶数)
	5	4	4	4	2	2	1		SpecialC	20002307-00	Backtracking (Than Two)
	<u> </u>	•	•	•	-	-	<u>'</u>	J	SpecialC Only	フロックと数字の配置から問題作成	Backtraking(C Solution)
									number of solution :	=10:0分 1秒904	Backtracking Solutions)

のように、これは10個の別解がある。別解を見ればわかるように、別解を除くために、1でプロ テクトしたつもりだったけど、フリーの1だったから、二つでサイズ2の領域ができてしまった。 他にも見落としがあったみたいです。

「フィルオミノ」の問題を人間が作る1つの方法は理解できたが、それなりに試行錯誤が必要 で、一本道で問題ができるわけではないみたいです。問題作成のエキスパートは何か特別なアイデ アまたは技法を持っているみたいですが、少し問題を解いてみたぐらいでは、「ひとりにしてくれ」 では簡単に見つかったアイデアや技法が、「フィルオミノ」では見えてきません。

「フィルオミノ」の問題を作るということは



のような局面の数字の1の所に、適当に数字をセットして、ユニークに解けるようにすれば良いで すから、虱潰しのプログラムをプログラミングすれば



のような問題を 26 秒ぐらいで作ってくれます。このように問題設定すれば探索空間の大きさが具 体的に分かります。即ち、探索空間は

$6^{18} = 101559956668416$

だけの組み合わせがありますので、勿論、上手く枝刈りする必要があります。解なしや複数解だ けでなく、難易度でも枝刈りしています。この方法でも、努力すれば、7×7の盤なら綺麗な問題 を作ってくれるようになりました。

もう一度プログラムを動かすと



のような問題を 21 秒ぐらいで作ってくれます。モチーフを変えると



のような問題を 39 秒ぐらいで作ってくれます。この解は論理的に考えて(即ち、私のプログラムで)

inpic net							 ・数値入力 	探索範囲の制限 縦 7 横 7	
2	1	5	5	5	3	3	5 数值入力 最	^{宇割合} 0.45 大の数字 6	
2	2	2	2	5	1	3	Reset	SpecialD	Backtrackir Two
5	5	5	25	5	5	5	First Step :	SpecialD Only	数字の点の
5	1			6	6	5	Simple Only	SpecialE Only	数字の点の配 問題作り
6	6	6	6	6	5	5	SpecialA Only	SpecialF SpecialF Only	数字の点の間間時作成(2
6	3		3	6	4	5	SpecialB SpecialB Only	ゴロッカレギヤぞの	「日見豊丁」「あっく」の 「新花(よ)(開設 「新見豊介作
		2		4	4	5	SpecialC 7	日本を教育の配	Backtrackin Than T
	1	_					SpecialC Only 問題作成:0分 39秒195	a.//*Drestatat11E/6%	Backtrakir Solutio

まで解けます。10×10の盤だと1から6までの数字しか使わなくても数字が 32 個なら探索空間は

 $6^{32} = 7958661109946400884391936$

だけの組み合わせがありますので、7×7の盤に比べ、10×10の盤だと

 $6^{14} = 78364164096$

倍、難しくなり絶望的で、論理的にブロックが決まる(即ち、プログラムでブロックが一意的に決 まる)数字のパターン(配置)を研究し、端の方から数字を決めるのではなく、そのパターンの部 分から数字を決めて、論理的に(プログラムで)大部分解けるようにしながら問題を構成していく など、更なる工夫しないと、「フィルオミノ」は数学的な議論で、難しいことが証明されているの で、現在私が作っているような単純な虱潰しのアルゴリズムでは、大きな盤の綺麗な問題は作れな いみたいです。単なるプログラミングの技術の問題ではなく、「フィルオミノ」の問題作成のエキ スパートがどのようにして問題を作っているかを本気になって研究する必要があります。

単純な方法でともかく問題が作れればいいと思ってプログラミングしてきて、取り扱うパズルの 数が増えるのを楽しみにしてきましたが、「四角に切れ」や「フィルオミノ」の問題作成を深く研 究してみて、いろんなことを経験し理解できたので、綺麗な問題が作れるように、今まで作ってき たプログラムを全部見直す必要があります。色々な外国語の勉強や将棋や囲碁などの趣味の他に、 センター試験の出題官をしている時(最後の日が東日本大震災の日で、地震の時には東京で電車に 乗っていました。何で電車が止まったままだろうと思っていると地震の揺れが来ました。電車が壊 れるのではないかというぐらい揺れましたが、乗客は皆平静でした。池袋から宿を取っていた渋谷 までてくてく歩きました。大変でした。)、飲み会に行く途中で、薩摩順吉先生がニコリの雑誌を 嬉しそうに買ってきたのを見てから鉛筆パズルにも興味を持ち、コンピュータで鉛筆パズルを解い たり、問題を作ったりする面白さを知ったので、当分、老後の暇つぶしができそうです。

一方、強いゲームのプログラミングには「機械学習」の理解がどうしても必要です。ゲームのプ ログラミングをするためには、「機械学習」や「深層学習」の本が沢山出版されるようになってき たので、理解できる本を探して、探索のアルゴリズムのように、もはや常識といえるように勉強 する必要があります。コンピュータ囲碁のプログラムで実現されていた「機械学習」の成果に違和 感を持っていた(何か違うなと感じていた)人は多いと思いますが、やっと本命の成果が「深層学 習」を使って実現され、欧州プロに勝ったとのニュースが世界中を駆け巡りました。実際に深層学 習を応用したプログラムを作るためには聞くところによると今は億単位のお金が必要(写真を見 て、猫を判別するプログラムのことです。これが出来るなら、囲碁の棋譜を学習させ、着手を予想 させることが出来ると誰でも思いつきます。私も昔、ニューラルネットワークを勉強した時、小さ い盤で、学習させてみましたが、パソコンでは全然ダメでした。「ゼロから作る Deep Learning」 を読んでみるとニューラルネットワークの基本も分かっていなかったことに気づきます!)みたい で、個人が趣味でする道楽では無理かも分かりませんが(聞くところによると深層学習の研究には 最低でも 300 万円のコンピュータが必要だそうです)、皆さんが中年になる頃には、誰でも使える 技術になっていると思います。最近、訓練済みのライブラリーが使えるようになってきて、色々試 せるようになっているみたいです。

数独の解法のプログラムをどうしても作りたければ、Java の本ですが、棚床弘樹著「鉛筆パズ ルゲームプログラミング ナンバープレス、お絵かきパズル、ナンバークロスワードのアルゴリ ズム」ソフトバンククリエイティブ 2007年を図書館で探すか古本屋で探して読んで下さい。 残念ながら絶版になっています。この本を最初に読んだ時には、問題作成に興味がなかったので気 にも留めず飛ばし読みしていましたが、書評に問題の作り方も書いているとあるのに最近気づき、 この本は2冊持っているはずですが、高知市のマンションと佐川町の実家の4か所ある書棚のど こにしまったか行方不明なので、2018 年4月4日、古本屋で三冊目を手に入れ読んでみると確か にナンバープレース(数独)の問題の作り方も書いてありました。最初からこの本を隅から隅まで 読んでおけば、余計な努力が必要なかったような気もしますが、他人がやったことを単になぞるの ではなく、自分で色々考えて、思いついたことを片っ端からプログラミングしてみることは勉強に なります。また、Rubyの本で良ければ、 David Flanagan まつもと ゆきひろ著「プログラミン グ言語 Ruby」O'REILLY にも、Ruby による数独を解くためのプログラムが載っています。イン ターネットにも色々な情報があります。数独以外のパズルの解法のプログラミングの例は、「ひと りにしてくれ」の解法解析の文書も株式会社ニコリさまの許可が下りましたので同じところにアッ プしていますので、それを参照してください。問題作成のエキスパートがどんな技法を駆使して 問題を作っているか分かります。 鉛筆パズルとパソコンは相性がいいです。パソコン教室と言え ば、今は、Excel、 Word、デジカメが御三家の世の中で(こんなものが何で面白いのだろうと私 は思いますが)、昔パソコン教室で BASIC を勉強したけど理解できなかったという話もよく聞き、 アマチュアのプログラマーは少なくなっているように思いますが、C++ が難しければ(配列の範 囲外に飛び出したことをコンピュータが教えてくれないので、コンピュータが異常停止したら、ま ずそれを疑って、原因を探れば良いだけで、C++ がことさら難しいわけではないと思いますが)、 Scratch などの間違いを起こしにくく設計されているプログラミング言語もあります。プログラム を組んでパソコンを思い通りに動かすことは楽しいことです。複雑なグログラムを作るとバグ取り ばかりで、なかなかそうは上手くいきませんが。是非、やってみてください。