

C 言語、C++ 言語

C 言語と C++ 言語

- [C 言語](#)は 1972 年から始まった古い言語だが、未だに現役。← J/ψ の発見は 1974 年
- [C++ 言語](#)は C 言語から派生した言語。1983 年に誕生した。← W/Z 発見の年, TDL 開園の年
- どちらも古い言語だが、規格を改定し新しい機能の取り込みや整備が続けられている。
- 入門書でちゃんと学習することをおすすめします。
 - C 言語の入門書：基本的な文法などを学ぶにはこっちがいい
 - C++ の入門書：大抵は C 言語の入門書程度の知識を前提としているので、こっちから始めるのはおすすめしない
- 糠塚の資料は[こちら](#)

C/C++ で勉強しておきたいこと

別にこの本を宣伝しているわけではないが…

- 第1章 まずは慣れよう
- 第2章 演算と型
- 第3章 プログラムの流れの分岐
- 第4章 プログラムの流れの繰り返し
- 第5章 配列
- 第6章 関数
- 第7章 基本型
- 第8章 いろいろなプログラムを作ってみよう
- 第9章 文字列の基本
- 第10章 ポインタ
- 第11章 文字列とポインタ
- 第12章 構造体
- 第13章 ファイル処理



- 第1章 画面への出力とキーボードからの入力
- 第2章 プログラムの流れの分岐
- 第3章 プログラムの流れの繰り返し
- 第4章 基本的なデータ型
- 第5章 配列
- 第6章 関数の基本
- 第7章 ポインタ
- 第8章 文字列とポインタ
- 第9章 関数の応用
- 第10章 クラスの基本
- 第11章 単純なクラスの作成
- 第12章 変換関数と演算子関数
- 第13章 静的メンバ
- 第14章 配列クラスで学ぶクラス的设计



C/C++ で勉強しておきたいこと

別にこの本を宣伝しているわけではないが…



第1章	まずは慣れよう
第2章	演算と型
第3章	プログラムの流れの分岐
第4章	プログラムの流れの繰り返し
第5章	配列
第6章	関数
第7章	基本型
第8章	いろいろなプログラムを作ってみよう
第9章	文字列の基本
第10章	ポインタ
第11章	文字列とポインタ
第12章	構造体
第13章	ファイル処理

100 ページ
くらい

このくらいは理解して
いないと始まらない

第1章	画面への出力とキーボードからの入力
第2章	プログラムの流れの分岐
第3章	プログラムの流れの繰り返し
第4章	基本的なデータ型
第5章	配列
第6章	関数の基本
第7章	ポインタ
第8章	文字列とポインタ
第9章	関数の応用
第10章	クラスの基本
第11章	単純なクラスの作成
第12章	変換関数と演算子関数
第13章	静的メンバ
第14章	配列クラスで学ぶクラスの設計

200 ページ
くらい



C/C++ で勉強しておきたいこと

別にこの本を宣伝しているわけではないが...



第1章	まずは慣れよう
第2章	演算と型
第3章	プログラムの流れの分岐
第4章	プログラムの流れの繰り返し
第5章	配列
第6章	関数
第7章	基本型
第8章	いろいろなプログラムを作ってみよう
第9章	文字列の基本
第10章	ポインタ
第11章	文字列とポインタ
第12章	構造体
第13章	ファイル処理

100 ページ
くらい

50 ページくらい

30 ページくらい

このくらいは理解して
いないと始まらない

初心者の難関

第1章	画面への出力とキーボードからの入力
第2章	プログラムの流れの分岐
第3章	プログラムの流れの繰り返し
第4章	基本的なデータ型
第5章	配列
第6章	関数の基本
第7章	ポインタ
第8章	文字列とポインタ
第9章	関数の応用
第10章	クラスの基本
第11章	単純なクラスの作成
第12章	変換関数と演算子関数
第13章	静的メンバ
第14章	配列クラスで学ぶクラス的设计

200 ページ
くらい

100 ページ
くらい



C/C++ で勉強しておきたいこと

別にこの本を宣伝しているわけではないが...



第1章	まずは慣れよう	
第2章	演算と型	100 ページ くらい
第3章	プログラムの流れの分岐	
第4章	プログラムの流れの繰り返し	
第5章	配列	
第6章	関数	
第7章	基本型	50 ページくらい
第8章	いろいろなプログラムを作ってみよう	
第9章	文字列の基本	30 ページくらい
第10章	ポインタ	
第11章	文字列とポインタ	
第12章	構造体	
第13章	ファイル処理	20 ページくらい

このくらいは理解していないと始まらない

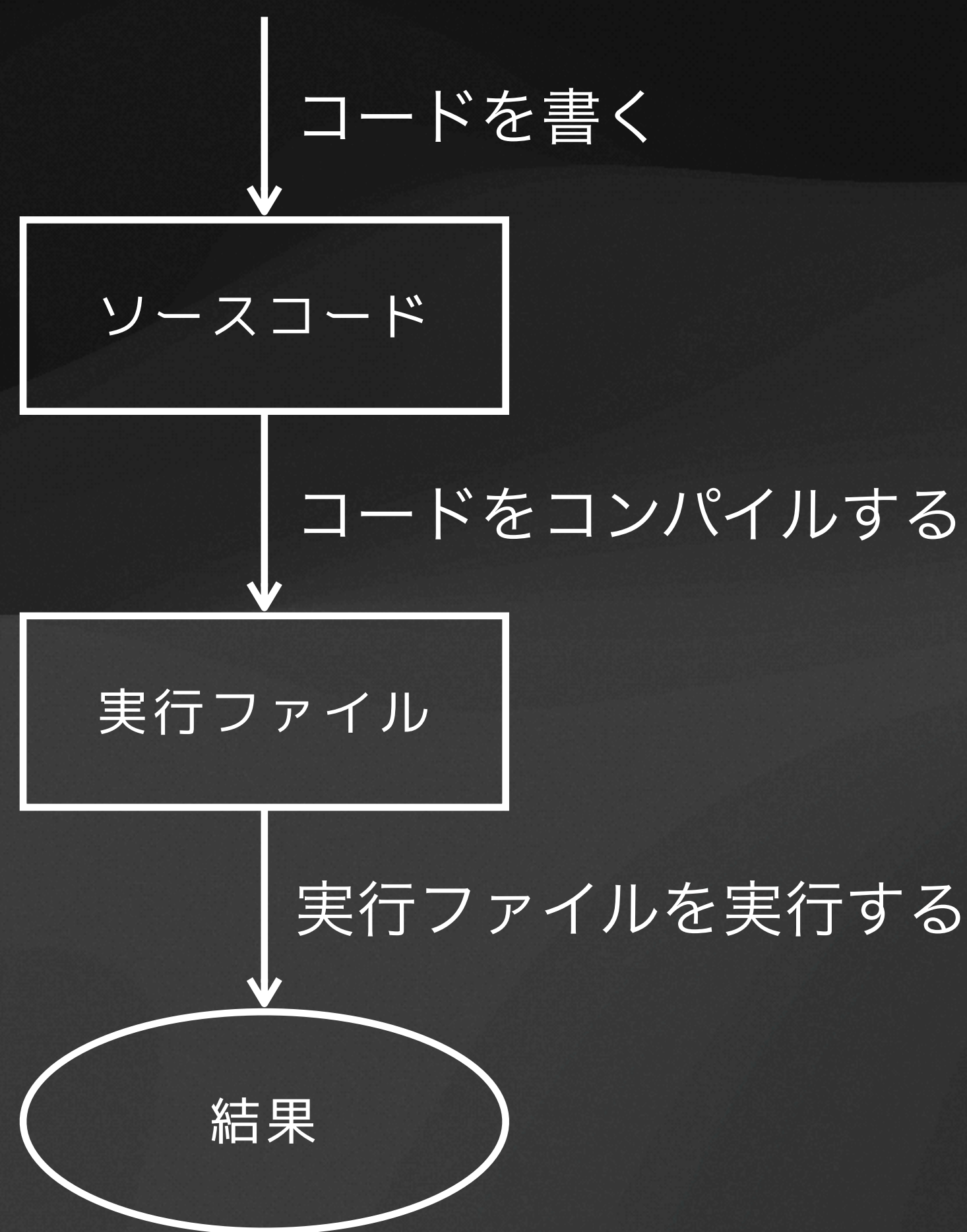
初心者の難関

Fun4All をちゃんと使うならこの理解が必要

第1章	画面への出力とキーボードからの入力	
第2章	プログラムの流れの分岐	
第3章	プログラムの流れの繰り返し	200 ページ くらい
第4章	基本的なデータ型	
第5章	配列	
第6章	関数の基本	100 ページ くらい
第7章	ポインタ	
第8章	文字列とポインタ	
第9章	関数の応用	
第10章	クラスの基本	
第11章	単純なクラスの作成	100 ページ くらい
第12章	変換関数と演算子関数	
第13章	静的メンバ	
第14章	配列クラスで学ぶクラス的设计	



C/C++ プログラミングの手順



1. プログラムのコードを書く
2. コードをコンパイルし、実行ファイルを作る
 - ・ 人はコードを読めるが、コンピューターはコードを直接四で理解することはできない。コンピューターの理解できる形式に変換すること（他にもやっていることはあるが）をコンパイルという。
3. 実行ファイルを実行する

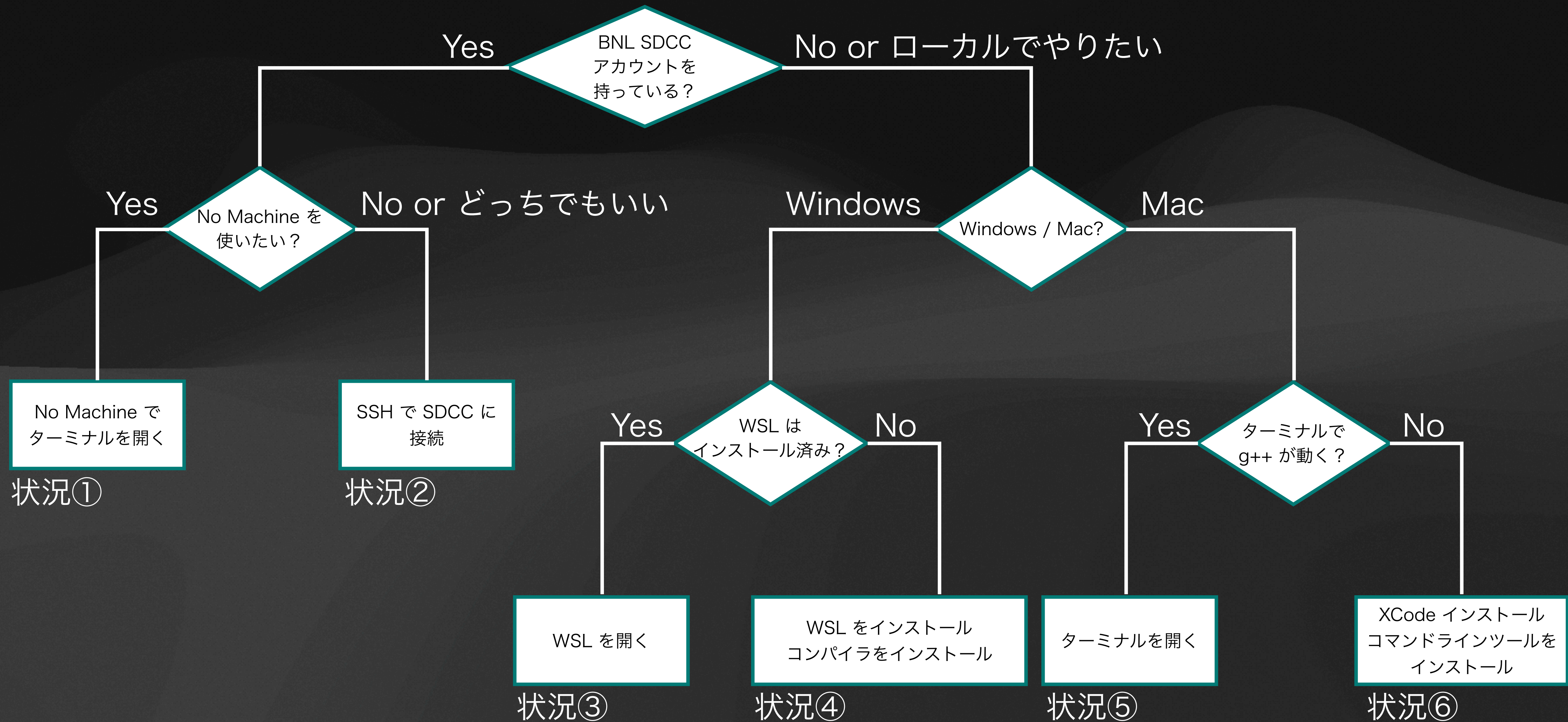
```
mail:answer genki$ █
```

例：プログレスバー

C/C++ プログラミングの準備 (環境構築)

- ・ 初心者にとって最大の壁はプログラミングができるように環境を構築すること・・・
- ・ みなさんはプログラミングできる状態だと思っっていますが、一応チェックしましょう

C/C++ プログラミングの準備：チェック



皆さんはどの状況ですか？Zoom のチャットで教えて下さい。

C/C++ プログラミングの準備：状況④の人

1. WSL2 のインストール：<https://learn.microsoft.com/ja-jp/windows/wsl/install>

2. コンパイラのインストールはターミナル上で

```
sudo apt install build-essential
```

を実行すればいいはず

C/C++ プログラミングの準備：状況⑥の人

1. Xcode のインストール



2. ターミナルで

`xcode-select --install`
を実行する



やってみよう 1 : Hello World

- ・ ターミナルに
Hello, world!
と表示する C++ プログラムを作成し、コンパイル、実行してみましよう。
- ・ 自信のある人は何も見ないで書いてみましょう。
- ・ 自信のない人は次のページへ

やってみよう 1 : Hello World 回答例

https://github.com/nukazuka/INTT_Fun4All_Tutorial/blob/main/ver2024/cpp_basics/answers/sample2.cc

コード

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "Hello, world!" << std::endl;
6
7     return 0;
8 }
```

コンパイル

```
[nukazuka 19:05:31 answers] $ g++ sample2.cc
```

実行

```
[nukazuka 19:05:43 answers] $ ./a.out
Hello, world!
```


やってみよう 2 : for 文と if 文

例 (for)

for 文

書き方 (これ以外にもある) :

```
for ( 初期化 ; 継続条件 ; 次のループに入るときにする処理 )
{
    やることを書く
}
```

簡単な例 :

```
for ( int i = 0; i<10; i++ )
{
    cout << i << endl;
}
```

初期化 : int 型変数 i を定義し、初期値 0 を i に代入する
継続条件 : i<10 が満たされる (この条件式が true を返す) 限り
次ループへの処理 : i の値に 1 を加える

ループ内でやること : 変数 i をターミナルに表示する

```
[genki 02:23:31 ~] $ cat > sample_for.cc
#include <iostream>

int main()
{
    for( int i=0; i<10; i++ )
    {
        std::cout << i << std::endl;
    }

    return 0;
}
^C
[genki 02:24:49 ~] $ g++ sample_for.cc
[genki 02:25:13 ~] $ ./a.out
0
1
2
3
4
5
6
7
8
9
```


やってみよう2 : for 文と if 文

if 文

例 (if)

書き方① :

```
if ( 条件式 )
{
    やることを書く
}
```

条件式が真 (true) のときのみ
実行される

書き方② :

```
if ( 条件式 )
{
    やることを書く
}
else
{
    やることを書く
}
```

条件式が偽 (false) のときのみ
実行される

書き方③ :

```
if ( 条件式 )
{
    やることを書く
}
else if( 条件式 )
{
    やることを書く
}
else
{
    やることを書く
}
```

簡単な例 :

```
int num = 5;
if ( num < 10 )
{
    cout << num << endl;
}
```

条件 : num < 10

条件が真のときやること : 変数 num をターミナルに表示する

```
[genki 02:25:15 ~] $ cat > sample_if.cc
#include <iostream>

int main()
{
    int num = 5;
    if( num < 10 )
    {
        std::cout << num << std::endl;
    }

    return 0;
}
^C
[genki 02:34:28 ~] $ g++ sample_if.cc
[genki 02:34:53 ~] $ ./a.out
5
```


やってみよう 2 : for 文と if 文

1. 例 (for) と例 (if) を自分で書いてコンパイルし、実行してみよう
2. 時間があまった人は[ここ](#)の課題 0~3 をやってみましょう
3. それでも時間があまった人は[ここ](#)の課題 1~3 をやってみましょう

```
mail:answer genki$ █
```

最終課題はこんな感じ