新型コロナウイルス感染対策の今一度の徹底を！

大阪大学は、学生・教職員全員が以下のルールを守ることで講義や研究、課外活動の制限を最小限に留めています！

絶対に守ってほしいこと

- **マスクルールの徹底**（屋内では必ず！屋外での会話も！）
- **個食の徹底**
  飲み会、カラオケ、友人宅での団らん等で感染が広まっています

- 体調不良の場合は登校せず、**かかりつけ医や保健所に相談**すること
- 保健所の**指示は法定**です。**必ず従う**こと
  自宅待機の指示やPCR検査の受診など、**最優先で従って**ください
- 保健所とのやり取りは、**直ちに**大学へ連絡すること
  基本は所属部局の教務係、夜間・休日の場合は covid-19@dash.osaka-u.ac.jp

マイハンダイアプリから送れます！

演習の実施形態

- 本演習はハイブリッド形式とする。
  「対面」および「後日動画配信」とする。

- 今学期は出席を取らず、レポートのみで成績評価する。

※新型コロナウィルス感染症の影響により、実施形態が
　変わる可能性がある。実施形態変更の場合はKOANにて通知。

コンピュータサイエンスとプログラミング
第１回演習　ガイダンス

2021年04月15日(木)

受講生の確認

◇ 平成29年度以降入学者

電子情報工学科情報通信工学科目(コース未分属)
電子情報工学科情報通信工学科目
（通信工学 or 情報システム工学コース）

◇ 平成28年度以前入学者

電子情報工学科　情報通信工学科目
（情報通信工学コース、通信工学クラス or 情報システム工学クラス）

◇ 工学部電子情報工学科以外の受講生の有無の確認

成績評価とKOANでの履修登録について

① 本科目は**必修科目(学科コア科目)**である.

② 講義部分(木曜1限開講) 50%, 演習部分(木曜3限開講) 50%の比率にて成績を評価する.

③ 演習部分については, ~~出席点 および~~ **レポート点**で成績を評価する.

④ コンピュータサイエンスとプログラミングI(春学期)とII(夏学期)に分かれています. **I・II 両方の履修登録**が必要.

⑤ A組とB組で時間割コードは異なる. 履修登録時に誤って登録しないよう注意すること.

再履修学生の受講クラスについて

情報システム工学専門実験第１部および
通信工学専門実験第１部(木曜3限～4限)の履修と
重複する場合に限り，受講クラスを下記の通りとする.

- 演習を木曜1限に実施する「コンピュータサイエンスとプログラミング(A組)」での，また，講義を金曜1限に実施する「コンピュータサイエンスとプログラミング(A組)」での受講を認める.
- A組とB組の時間割コードは異なる．履修登録時に時間割コードを誤って登録しないよう注意すること.

講義に関する基本情報

◇ 演習時間

- 木曜3限(13:30 ～ 15:00)

◇ 実施場所

- GSEコモン西棟2階 情報実習室 A室(U1W-211)およびB室(U1W-212)
- 受講者数を確認し，A室・B室のいずれかのみで実施できる場合は，どちらか一方の教室にて実施する.

◇ 担当教員およびTA担当学生

- 教員：三科 健，御堂 義博
- TA (Teaching Assistant):江木，川村，藤井，寺師，中尾

演習の実施形態と内容

◇ 実施形態

- 演習課題を実施し，それに対するレポートの提出.
- 講義時間内にレポートを提出できない場合，演習室の空き時間もしくは自身が所有するPCを用いて実施し，レポートを提出すること.
- 演習室のルールを遵守すること.
- 演習室を利用する他の授業時間中は利用しないこと.

◇ 講義内容(案)

① C言語によるプログラミング入門

② 基本的なデータ構造，優先度つき待ち行列

③ 2分探索木と平衡木

④ 整列の諸アルゴリズム

講義スケジュール(予定)

	回	講義日	内容
I	第01回	4/15	ガイダンス
	第02回	4/22	関数の定義と再帰呼び出し
	第03回	5/6	モンテカルロシミュレーション
	第04回	5/13	ポインタ
	第05回	5/20	構造体１
	第06回	5/27	構造体２
	第07回	6/3	連結リストの実現と操作
II	第08回	6/10	スタックの実現と操作
	第09回	6/17	２分木
	第10回	6/24	２分探索木
	第11回	7/1	待ち行列の実現と操作
	第12回	7/8	整列のアルゴリズム１
	第13回	7/15	整列のアルゴリズム２

教科書・参考書等(1)

◇教科書:「アルゴリズムとデータ構造(未来へつなぐデジタルシリーズ10)」

◇参考図書1(C言語を用いたプログラミングに関する書籍)

◇参考図書2(C言語でのアルゴリズムとデータ構造に関する書籍)

- 自身のスキルに応じた参考図書の購入を勧める.
- 上記の他, 必要に応じて, サイバーメディアセンター情報教育システムのホームページ等で必要な情報が掲載されていないか確認すること.

教科書・参考書等(2)

◇参考図書3(教育用計算機システムの利用方法に関する書籍)

サイバーメディアセンター 教育用計算機システム 利用の手引
大阪大学サイバーメディアセンター情報教育システムのリンク先よりPDFにて入手可能
http://www.ecs.cmc.osaka-u.ac.jp/wiki/

情報教育システム利用上の注意

- ログインに時間がかかる場合があります
  →教室に来たら、できるだけ早くログインしてください
- Windowsのユーザプロファイルが壊れ、ログインしてもツールバーなどが表示されない
  →TAもしくは教員に申し出てください
- デスクトップにデータを保存しない
  →必ず自分のドメイン下に保存
- 情報実習室以外のPCでプログラムを組む際は、文字コードに注意
  →必ずUTF-8で保存

講義情報

◇配布資料の電子ファイル，連絡事項は大阪大学CLE(授業支援システム)に掲載

https://cle.koan.osaka-u.ac.jp/

- KOANや大阪大学サイバーメディアセンター情報教育システムを通じてログインすることも可能.

質問について

◇質問用メールアドレス

csp-query@pn.comm.eng.osaka-u.ac.jp

・CLE経由で質問メールを送付しないで下さい。
TAメンバーに質問が届かず、回答ができないことがあります。
・できません、分かりません、動きません、といった丸投げする質問には回答しません。自分で何を試し、どういった状況でどのようなエラーが発生するのか具体的に記載してください。
※エラー発生時のソースコード(.cのファイル)やプリントスクリーンの画像ファイルを添付して送ってもらえると適切なアドバイスができます。
※氏名や宛名がないメールにも回答しません。

・質問の回答に時間がかかることがあります。課題に関する質問は早めに。
(締切前日に質問しても、締め切りまでに返事できるとは限りません。)

・デバグは基本的にエラーメッセージ等を見ながら自分で行いましょう。

プログラム実行までの過程

① プログラムの構想

▼

② プログラムリストの入力

エディタは何を使う？

▼

③ ファイル保存

ファイルはどこに保存される？

▼

④ コンパイル（エラーの場合，①または②へ）

コンパイラは何を使う？

▼

⑤ プログラム実行

プログラムの実行方法は？

① pwd（現在のディレクトリの確認）
② cd（ディレクトリの移動）

統合開発環境(IDE)の利用

- 統合開発環境を使うとプログラミングが大変楽になります.
  - エディタ、コンパイラ、デバッガなどが一つのGUIから利用できる。
  - Visual Studio: https://visualstudio.microsoft.com/ja/free-developer-offers/
  - JetBrains: https://www.jetbrains.com/ja-jp/cpp/
- 課題提出時には**gccでコンパイル・実行できる状態**で提出すること！

レポートに記載する内容

レポートとして記載する内容は下記の①～③とする.

① プログラミング方針

② 実行結果

③ プログラムを作成する上での困難だった点，工夫点

特に，③は理解度・オリジナリティを評価する上で重要なため，相応の文章および相応の内容を記載すること.

- 特に指示のない限り，演習課題の提出期限は課題提示1週間後の午前9時まで.
- 提出期限を過ぎても，各セメスターの最終提出日（最後の課題の提出期限）まで受け付ける.

レポート提出方法

1) 毎回のレポート課題はCLEを通じて提出すること.

2) プログラムリストは添付ファイルとして提出すること.

- ファイル名は必ず半角英数にて「XXXX-YYYY.c」とすること. XXXXは学籍番号下4桁, YYYYの部分は各回ごとに指定する.
- 指定したファイル名に従っていない場合には, 採点時にファイルを見つけることができず採点されない可能性があるので厳守すること.

(例)学籍番号87654321の学生が「YYYY」の部分を「kadai1-1」と指定された場合の添付ファイル名 ➔ 4321-kadai1-1.c

3) ①プログラミング方針, ②実行結果, ③プログラムを作成する上で困難だった点, 工夫点など, レポートに記載すべき事項については, CLEの課題提出画面の「テキスト情報」に記載すること.

- 上記指示に従っていない場合には, 採点時に記載事項①～③を見つけることができず採点されない可能性があるので厳守すること.

重要：課題提出時の注意点

1. レポートはCLEの**テキスト情報**に記入すること
   - メモ欄は読みにくく，見落とす可能性があります
2. コンパイル・実行に必要な**ソースはすべて**提出してください
   - 自作のヘッダファイルを作った人は，プログラム本体と一緒に必ず提出してください
   - gcc でのコンパイル・実行ができることを必ず確認して提出してください。
3. CLEのサーバトラブルなど不測の事態に備えて，余裕をもって課題を提出すること
   ※CLEがダウンしている，操作方法が分からない場合は**CLEのサポートに問い合わせてください**

レポート提出最終期限

- CSP I（第1回～第7回）の課題提出〆切

6/10 9:00

- CSP II （第8回～第13回）の課題提出〆切

7/22 9:00

演習点の評価基準

■ 各回の演習点は，出席点とレポート点により評価.

■ **出席点の 評価基準**:

- 欠席の場合，受講態度の悪い場合 ➔ 50～100%減点.

■ **レポート点 評価基準**:

- レポートの提出が遅れた場合 ➔ 30～70%減点.
- 指示通りに提出されていない場合（レポートに記載すべき事項①～③が記載されていない場合）➔ 0～100%減点.
- 複数の学生から類似レポートが提出された場合，当該学生全員（複製元となったレポートを提出した学生を含む）➔ 100%減点.

質問用メールアドレス

◇質問用メールアドレス

csp-query@pn.comm.eng.osaka-u.ac.jp

・CLE経由で質問メールを送付しないで下さい。
TAメンバーに質問が届かず、回答ができないことがあります。
・できません、分かりません、動きません、といった丸投げする質問には回答しません。自分で何を試し、どういった状況でどのようなエラーが発生するのか具体的に記載してください。
※エラー発生時のソースコード(.cのファイル)やプリントスクリーンの画像ファイルを添付して送ってもらえると適切なアドバイスができます。
※氏名や宛名がないメールにも回答しません。

・質問の回答に時間がかかることがあります。課題に関する質問は早めに。
(締切前日に質問しても、締め切りまでに返事できるとは限りません。)

・デバグは基本的にエラーメッセージ等を見ながら自分で行いましょう。

おかしい？と思ったら…

```
01 int main(){
02         int i,j;
03         for(i=1;i<6;i++){
04         for(j=1;j<6;j++){
05         printf("%d X %d = %d¥n",i,j,i*j);
06         }
07         printf("-------------------¥n");
08         }
09         return 0;
10 }
```

①エラーメッセージを読む。
エラーメッセージをコピペしてぐぐる。

②IDEを使う

- ブレークポイントを置いて、ステップ実行
- 変数にどんな値が格納されているのか確認できる

printfデバッグ（古い）

- プログラムの随所にprintfを仕込んで，必要な情報をダンプして調べる方法.
- 随所にprintf を埋め込みすぎると，ソースが見にくくなってしまう.

単純だけどよくあるバグ

- 全角アルファベット、スペースが入りこんでいる
- スペルミス
- 末尾のセミコロンがない
- プログラムの文字コードがUTF-8になっていない

エラーメッセージをよく読むこと！

例題1−1

【例題 1−1】

(1) 下記のプログラムの出力結果を予想しなさい.

(2) 下記のプログラムをエラーなくコンパイルできるよう完成させなさい.

(3) プログラムを実行し, 実行結果を確認しなさい.

```
int main(){
    int i,j;
    for(i=1;i<6;i++){
        for(j=1;j<6;j++){
        printf("%d X %d = %d¥n",i,j,i*j);
    }
        printf("--------------------¥n");
    }
    return 0;
}
```

課題1−1，課題1−2

【課題 1−1】

x を 9 以下の奇数とし，y を x 以下の奇数とする．$x+y$ の計算結果が下記のように表示されるプログラムを作成せよ．

```
9 + 1 = 10
9 + 3 = 12
9 + 5 = 14
9 + 7 = 16
9 + 9 = 18
--------------------
7 + 1 = 8
7 + 3 = 10
7 + 5 = 12
7 + 7 = 14
--------------------
```

```
5 + 1 = 6
5 + 3 = 8
5 + 5 = 10
--------------------
3 + 1 = 4
3 + 3 = 6
--------------------
1 + 1 = 2
--------------------
```

【課題 1−2】

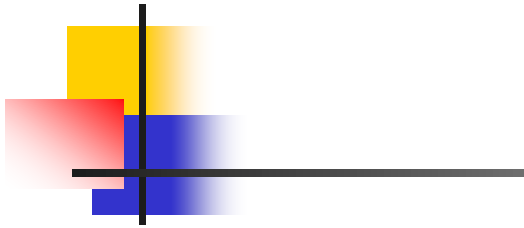
x を 7 以下の自然数とし，y を x 以下の自然数とする．mod(x, y)の計算結果が下記のように表示されるプログラムを作成せ

```
mod(1,1) = 0
-----------------------
mod(2,1) = 0
mod(2,2) = 0
-----------------------
mod(3,1) = 0
mod(3,2) = 1
mod(3,3) = 0
-----------------------
mod(4,1) = 0
mod(4,2) = 0
mod(4,3) = 1
mod(4,4) = 0
-----------------------
mod(5,1) = 0
mod(5,2) = 1
mod(5,3) = 2
mod(5,4) = 1
mod(5,5) = 0
```

```
-----------------------
mod(6,1) = 0
mod(6,2) = 0
mod(6,3) = 0
mod(6,4) = 2
mod(6,5) = 1
mod(6,6) = 0
-----------------------
mod(7,1) = 0
mod(7,2) = 1
mod(7,3) = 1
mod(7,4) = 3
mod(7,5) = 2
mod(7,6) = 1
mod(7,7) = 0
-----------------------
```

レポート課題

- 課題1-1，1-2を実施し，レポート課題として提出すること.
- 提出期限: 2021年04月22日 午前9時
- CLEで提出する際のファイル名(半角英数)は下記の通りとする.
  - 課題1-1の場合，XXXX-kadai1-1.c
  - 課題1-2の場合，XXXX-kadai1-2.c
  - XXXXの部分は学籍番号下4桁である.



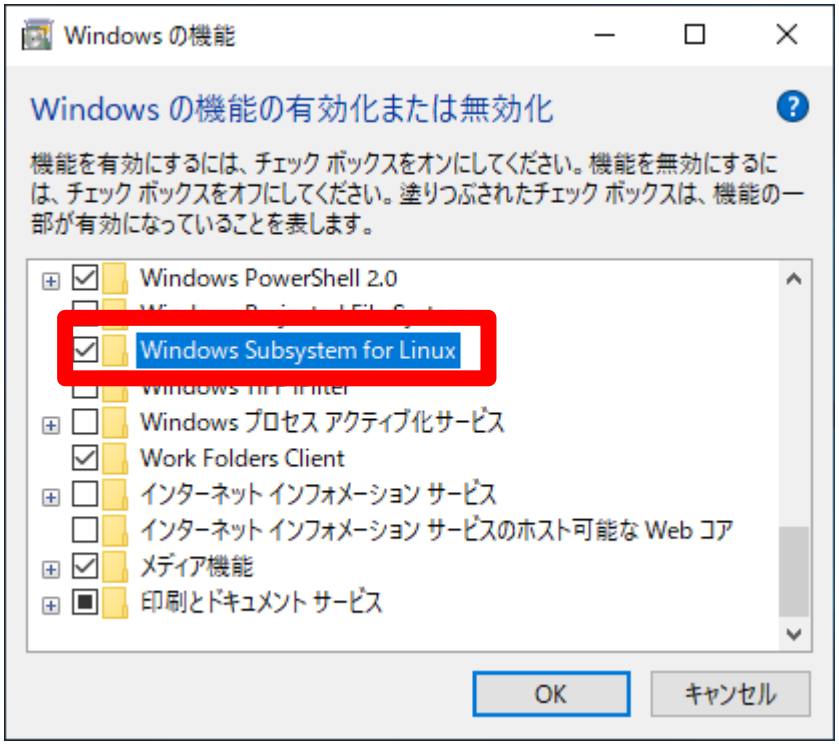
（参考）
所有PCに環境構築する場合

環境構築：Windows編

1. Windows Subsystem for Linux (WSL) のインストール
    1. [コントロールパネル]ー[プログラムと機能]ー[Windowsの機能]ダイアログを開く
    2. [Windowsの機能の有効化または無効化]をクリックしてダイアログを開き、そこで「Windows Subsystem for Linux」にチェックを入れる。
    3. PCを再起動する。

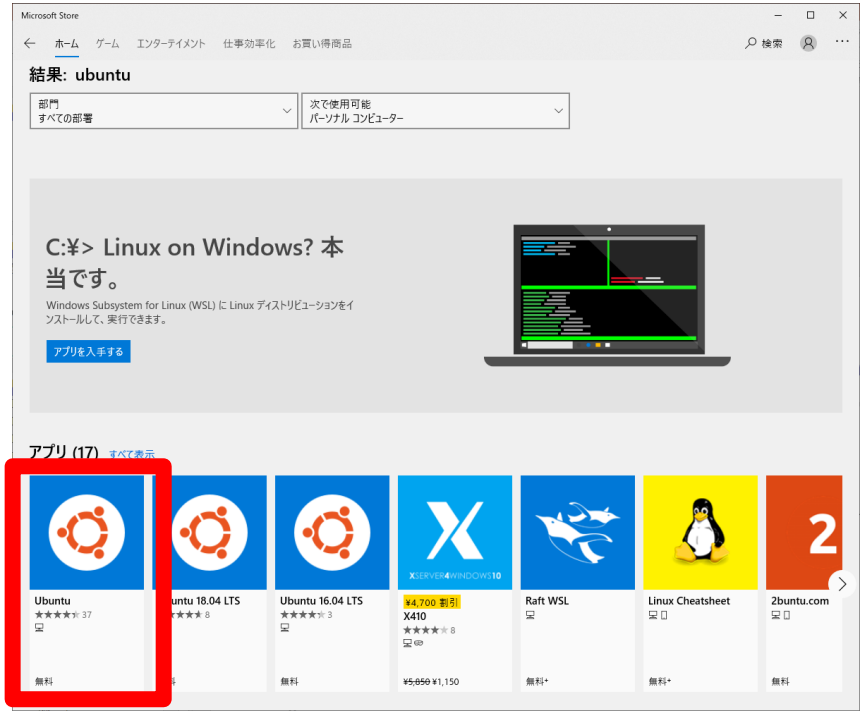
環境構築：Windows編

1. Windows Subsystem for Linux (WSL) のインストール



環境構築：Windows編

2. WSL上にUbuntu のインストール
    1. Microsoft Store を起動
    2. “Ubuntu” を検索、インストール

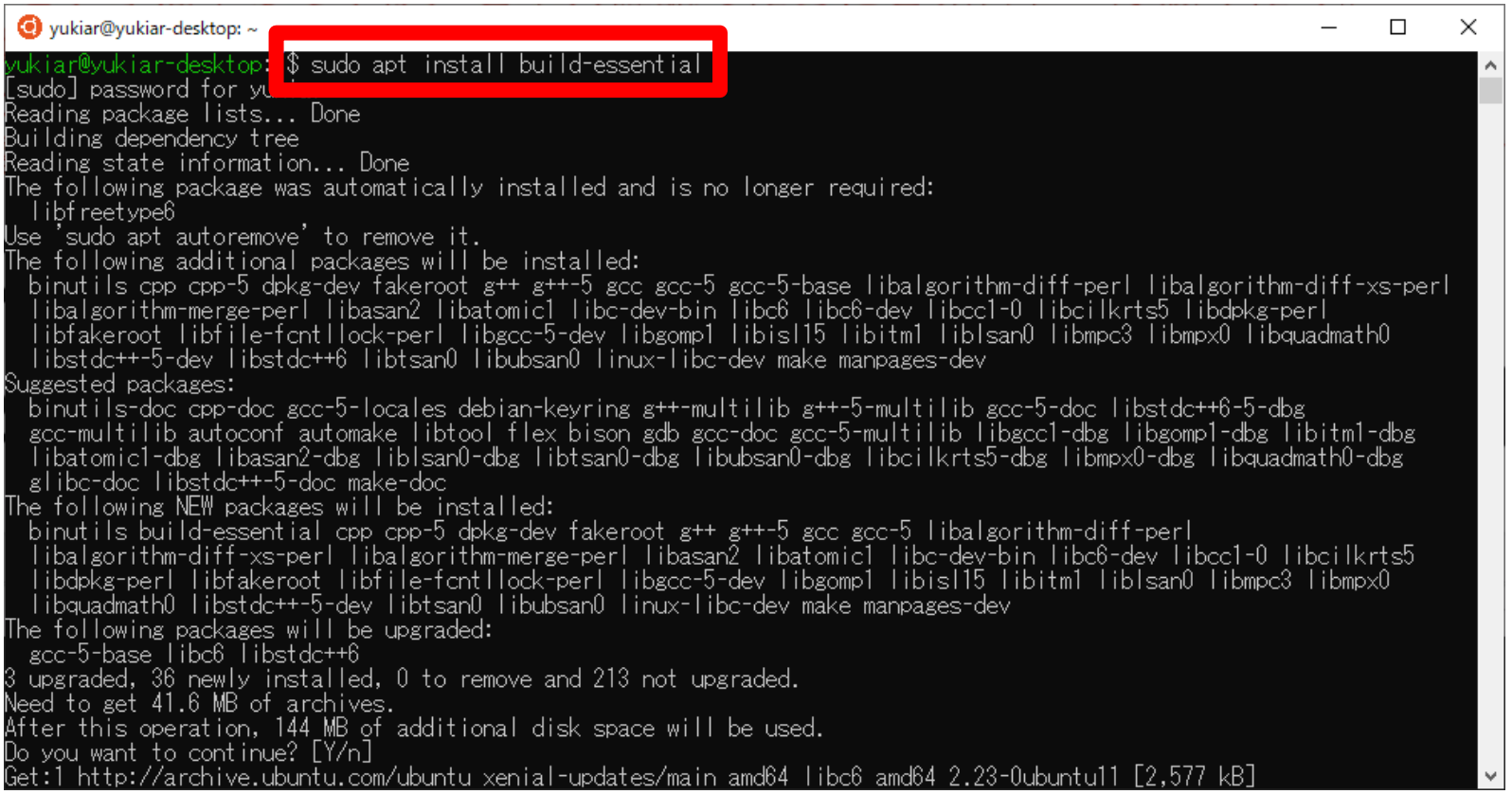


環境構築：Windows編

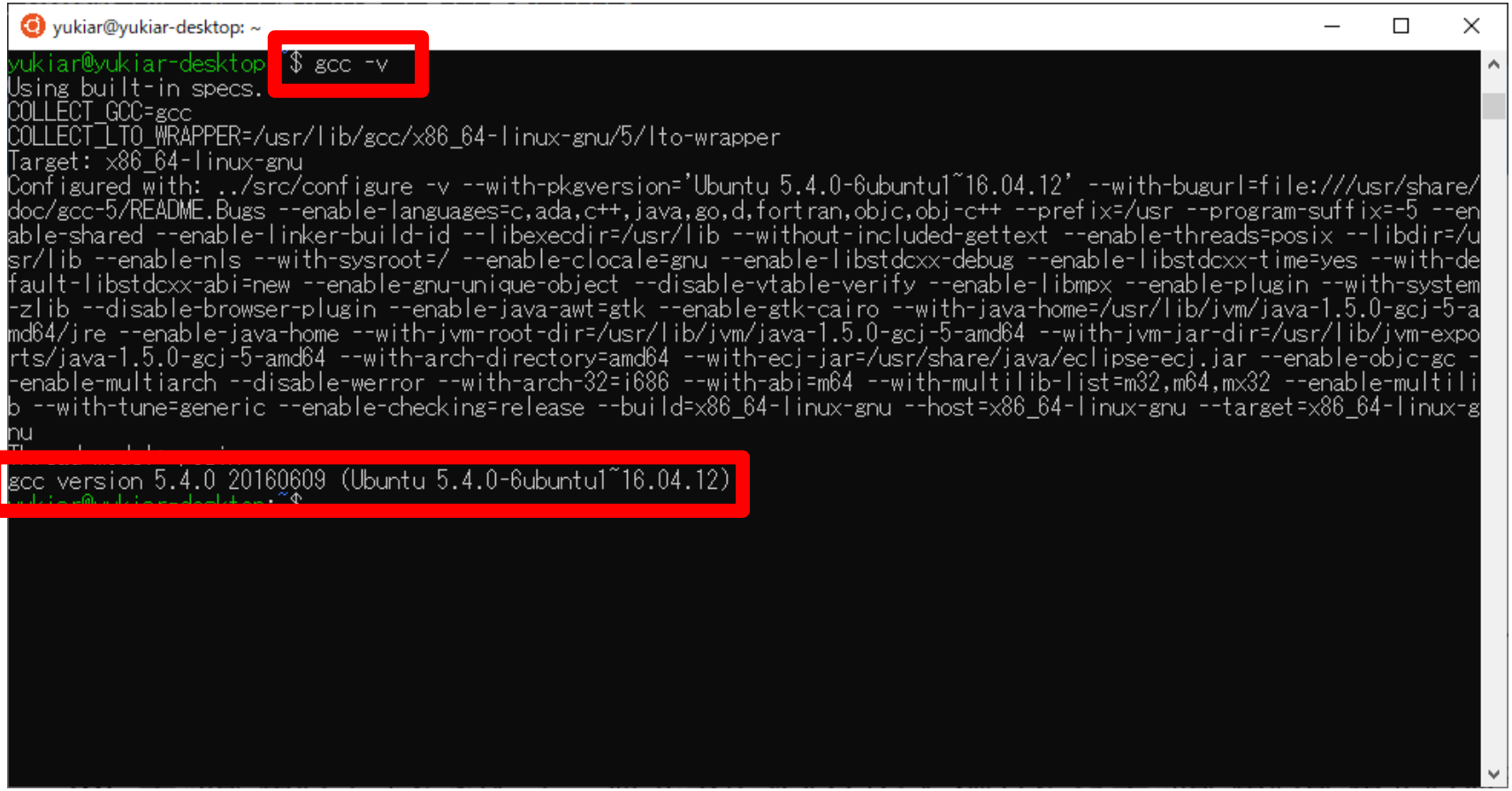
3. gcc のインストール
   1. Ubuntuを起動、ユーザ名・パスワードを設定
   2. `$sudo apt-get update` を実行
   3. `$sudo apt install build-essential` を実行
   4. `$gcc -v` を実行して、バージョンが表示されれば成功

詳細は以下のページを参照
https://www.atmarkit.co.jp/ait/articles/1903/18/news031.html

環境構築：Windows編



環境構築：Windows編



環境構築：Mac 編

- gcc のインストール
  1. `$ brew install gcc` を実行
  2. `$ gcc -v` を実行し、gcc のバージョンが表示されれば成功

詳細は以下のページを参照
https://qiita.com/wawawa/items/50c2c612b0937f28d92b

環境構築に関するQ&A

Q: gccがうまくインストールできない。

A: インストールに失敗する場合があります。

```
$ sudo apt-get upgrade
$ sudo apt install gcc
```

(https://www.yokoyan.net/entry/2019/01/25/181500 より)
を試してみてください。

それでもうまくいかなかった場合、

```
$ sudo apt update
$ sudo apt upgrade
```

を行ってみてください。

環境構築に関するQ&A

Q: gccでコンパイルできず、”gcc: error: XXX.c: No such file or directory”と
　表示される。

A: コンパイルできない原因は表示の通り，「ファイルにOO.cがない」ことです。
コンパイルするためにはファイルの保存されているディレクトリに移動しなくてはなりません（おそらくOO.c をCドライブのドキュメントなどに保存していると思います）。

WSLはやや特殊で、Cドライブに移動するには
$ cd ../../
$ cd mnt/c
とする必要があります。
※https://www.atmarkit.co.jp/ait/articles/1806/08/news042.html
　より「マウントされたドライブの一覧を確認する」

ファイルのあるディレクトリに移動後，gcc を実行すればコンパイルできると思います。

他にも、ファイル形式（xxx.c）や、アクセス権限設定にも注意！

環境構築に関するQ&A

Q: gccの後、どうすればよいか分からない。
”cannot open output file a.out”と表示される。

A: gccでコンパイルができた場合、実行ファイル(a.out) が生成されます。
プログラムを実行するには
$ ./a.out
とする必要があります（参考サイト：https://webkaru.net/clang/linux-gcc-compile/ ）。
もし同様のエラーメッセージがでるようであれば、一度実行ファイルを削除してから再びコンパイルしてみてください。
実行ファイル名を任意に設定することもできるので調べてみると良いでしょう。

環境構築に関するQ&A

Q: visual studio2019をダウンロードしようと思ったのですが、community, professional, enterprise の3つがあるうち、どれをダウンロードすれば良いでしょうか。

またダウンロードしたvisual studio2019の使い方について解説しているサイト等有れば教えていただきたいです。

A: Visual Studio Community以外は有料ですが、

本演習は無料のCommunityで対応可能です。。

使い方について解説しているサイトは多くあるとは思いますが、

公式のドキュメントもあるので、そちらを参考にしてみてください。

https://docs.microsoft.com/ja-jp/visualstudio/windows/?view=vs-2019