

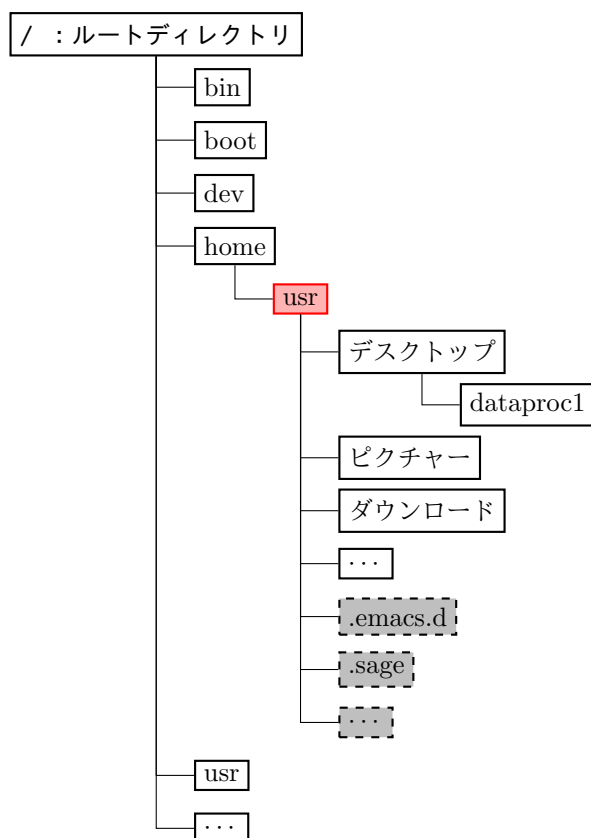
情報処理 I (2019 年度) 第 2 回目

1 Linux の使い方

今回は MathLibre の起動 USB メモリを作成し、それを講義用にカスタマイズしました。以下では、MathLibre を USB メモリから起動したものとして話を進めます。

1.1 Linux システムのファイル・ディレクトリ構造

コンピューター上のデータはすべてファイルという形で保存されています。すべてのファイルはディレクトリに入っています。ディレクトリのことをフォルダということもあります。Linux システムのディレクトリ構成は Windows や MacOS とは異なります。MathLibre のディレクトリ構成は以下のようになっています：



ディレクトリは辞書的で入れ子の構造をしており、階層の一番上にあるディレクトリはルートディレクトリと呼ばれ記号『/』で表されます。この MathLibre ではあなたのユーザー名は **usr** です。このコンピューターで作成する個人的なファイルはすべて **usr** 以下のディレクトリに保存されます。ホームディレクトリは記号『~』で表します。

ディレクトリやファイルはファイルマネージャーや端末などから閲覧・編集することができます。

ここでは端末からファイルやディレクトリを閲覧する操作については前回のプリントを参照してください。前回行った作業ではデスクトップに `dataproc1` というディレクトリを作成しました。またホームディレクトリに `Desktop` というシンボリックリンクを作成しました。端末を起動するとホームディレクトリにいますが、ここからデスクトップにある `dataproc1` に移動するには次のようにします。

```
1 | user@debian:~$ cd Desktop/dataproc1
```

```
2 | user@debian:~/デスクトップ/dataproc1$
```

ここからのファイルを作ってみましょう。

```
1 | user@debian:~/デスクトップ/dataproc1$ touch abcde.txt
```

```
2 | user@debian:~/デスクトップ/dataproc1$
```

するとディレクトリ dataproc1 に abcde.txt というファイルが作成されます。このファイルのディレクトリも含めたファイル名は

```
~/デスクトップ/dataproc1/abcde.txt
```

となります。この講義で作成するファイルはデスクトップにある dataproc1 に保存することにします。

ディレクトリの移動を行うときには、以下で説明する Tab キーによる補完を使うのが便利です。

1.1.1 端末の機能：Tab キーによる補完機能、履歴

ターミナルで様々なコマンドを実行するときには、一字一句正確に入力しなければならないのですが、Tab キー（キーボードの左側にある）を使って手間を省くことができます。以下では#以下はコメントなので、入力する必要はありません。ホームディレクトリで次のように入力してから Tab キーを押してみましょう：

```
1 | user@debian:~$ cd De # Tabキーを押す
2 | user@debian:~$ cd Desktop # Enterを押す
3 | user@debian:~/Desktop$
```

すると 2 行目のように自動的に残りの部分が補完されます。一般的にターミナルでコマンドの入力中に Tab キーを押すと、以降の選択肢が一つしか無い場合にはそれが補完され、複数の選択肢がある場合には、その候補が表示されます。例えば to に続いて Tab キーを押すと

```
1 | user@debian:~$ to # ここでTabキーを押す
2 | toe togeomview top touch
3 | user@debian:~$ to
```

のように、to から始まるコマンドのリストが表示されます。

また、端末で入力した履歴はキーボードの上・下のキーで見ることができます。同じ作業を繰り返す場合には、履歴から実行して手間を省きましょう。

1.2 ドットファイル (隠しファイル)

ホームディレクトリからコマンド ls を入力してもう一度中身を見てみましょう：

```
1 | user@debian:~$ ls
2 | Desktop テンプレート ドキュメント 音楽 公開
3 | ...省略...
4 | user@debian:~$
```

ファイルやディレクトリの種類の一つにドットファイル (隠しファイル) があって、それは通常では見えないうろなっています。システムファイルや設定ファイルなど日常的には必要がないファイルはドットファイル (隠しファイル) になっています。隠しファイルを表示するには次のようにします：

```
1 | user@debian:~$ ls -a # オプション -a で隠しファイルなども表示
```

2	.	.cache	.local	.uim.d	テンプレート
3	..	.config	.mozc	.wicd	デスクトップ
4	...省略...				

ファイルやディレクトリの先頭にドット「.」が付いているものは隠しファイル（またはディレクトリ）です。Linux では、設定ファイルがホームディレクトリ以下に隠しファイルとして作られていることがよくあります。

2 Emacs : 高機能テキストエディタ

テキストファイルの作成・編集を行うソフトウェアをテキストエディタといいます。Windows の場合はメモ帳 (notepad), MathLibre では Leafpad が標準的なテキストエディタですが、これらには最低限の機能しかありません。この講義では Emacs という高機能エディタを使用します。Emacs はキーバインディングが Windows の標準とは異なるので、使い始めは違和感がありますが、練習を積むことで効率よくテキスト編集ができるようになります。

2.1 Emacs の設定

前回行った操作で Emacs を講義用にカスタマイズしました。特に Ricty Diminished というプログラミング用のフォントがインストールしています。これは、見た目もよく 0(オー)と 0(ゼロ)の区別が明確で、全角空白に見えるようにしてあるのでプログラミングに最適のフォントです。Emacs はフリーソフトウェアなので Windows, MacOS でも誰でも無料で使用することができます。自宅の PC で講義と同じ設定を行うには講義用の Emacs の設定ファイル

<http://math.shinshu-u.ac.jp/~isasaki/classes/2019dp1/files/init.el>

を参考にしてください。

2.2 Emacs の起動

それでは、Emacs を起動してみましょう。Emacs は端末から

```
1 | $ emacs &
```

とすることで起動できます。また編集するファイル名を指定して起動することもできます。例えば abcde.txt を編集したい場合

```
1 | $ emacs abcde.txt &
```

とします。（注意）端末から Emacs を起動する際に単に emacs と入力して起動すると端末が使えなくなってしまいますが、emacs & という具合にアンドを付けて起動すると、端末を停止させないで emacs を使うことができます。このことは他のプログラムでも同じです。

Emacs を起動して、ディレクトリ dataproc1 に typetest.txt というファイルを作ってみましょう。

```
1 | $ cd ~/Desktop/dataproc1      # ディレクトリを移動
2 | $ emacs typetest.txt &       # Emacsでファイルを開く
```

次に適当に文字を入力してからファイルを保存します：

ファイルの保存 [Save] は [CTRL]+x [CTRL]+s で行います。

ファイルが保存されると画面下の方に Wrote /home/usr/Desktop/dataproc1/typetest.txt というメッセージがでます。デスクトップのフォルダ dataproc1 に typetest.txt が作られていれば成功です。ダブルクリックなどして中身を確認してみましょう。

Emacs の終了は [CTRL]+x [CTRL]+c で行います。

2.3 Emacs の操作

Emacs では [CTRL] キーを多用します。[CTRL] を押しながら g を押すという操作を C-g と略記します。よく使う機能を以下の表にまとめました。

コマンド	動作
C-g	コマンドを中断する
C-x C-w	名前を付けてファイルを保存
C-x C-s	コマンドを中断する
C-z	作業の取り消し, アンドゥ
C-s	前方検索 (検索から抜けるには C-g)
C-r	逆方向の検索
C-x C-c	Emacs の終了

コピー・ペーストは Windows の標準とは異なるので注意して下さい。Emacs では「マーク」というものを置いて、コピーやカットを行います。マークとカーソルの間の領域をリージョンといいます。

- C-SPC : マークセット (SPC はスペースキー)
- C-w : リージョンをカット
- Alt-w : リージョンをコピー
- C-y : カット (コピー) した文章をペースト

カーソルの移動は

C-f	右へ移動 (→)
C-b	左へ移動 (←)
C-p	上へ移動 (↑)
C-n	下へ移動 (↓)
C-a	行頭へ移動
C-e	行末へ移動

です。その他によく使う機能は

C-t	直前の 2 つの文字を入れ換え
C-k	カーソルから行末までをカット

です。例えば、『()aiueo』の『)』の部分にカーソルをおいてから C-t を連打してみると便利さが少しわかります。

リージョンを選択してから

C+SPC+c C+SPC+c

でマルチカーソルになります。複数行をコメントアウトするときに、リージョンを選択してマルチカーソルに

してから C-a で行頭に移動してコメントアウトを入力するのが便利です。

その他, Emacs の機能については Emacs のチュートリアルを見て下さい。チュートリアルは Help の一番上にあります。

3 文章作成ソフト・TEX

3.1 TEX とは？

TEX とは無料・オープンソースの文章作成ソフトです。日本では TEX をテフと発音します（テックと呼ぶ人もいます）。TEX に様々なマクロを組み込んで作業を自動化したものが L^ATEX (ラテフ) です。いまでは TEX といえばほとんどが L^ATEX の事を意味します。最近の数学書の多くが L^ATEX で作られています。この講義のプリントもすべて L^ATEX で作成しています。数学や物理などの科学論文の多くは L^ATEX で書いたものを投稿します。L^ATEX は誰でも無料で自分のコンピューターにインストールすることができます。

3.2 Windows で L^ATEX を使うには

インストール DVD 等の付属している L^ATEX の解説書を買ってきて、付属メディアからインストールするという方法が手軽です。また、様々な人が L^ATEX の便利なインストーラを作って公開しています。例えば東京大学の阿部紀行さんが作成した『TEX インストーラ』を使えば、クリックするだけで TEX をインストールすることができます。阿部紀行さんが公開している TEX インストーラの Web ページはこちらです：

<http://www.ms.u-tokyo.ac.jp/~abenori/index.html>

3.3 L^ATEX の使い方の流れ

TEX の文章作成の手順は次のようになっています：

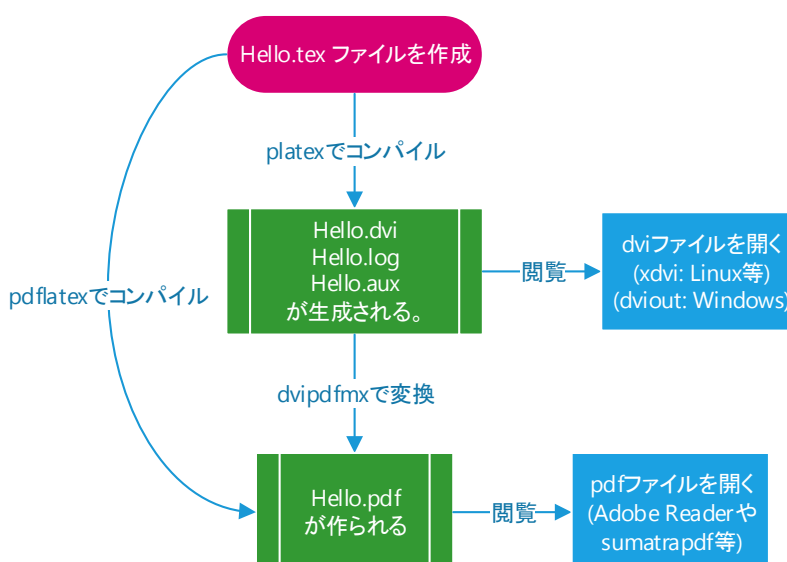


図 1 L^ATEX での文章作成

現在は pdflatex は日本語処理でエラーを起こすので使わないほうがよいでしょう。

また上の処理を一括で行うことができる TeXmaker, TeXstudio, TeXworks といった統合環境があります。

3.4 TeX 文章の作成

Emacs を起動して、dataproc1 に次の TeX ファイルを作ってみましょう。

```
1 user@debian:~$ cd Desktop/dataproc1 # ディレクトリを移動
2 user@debian:~/Desktop/dataproc1 $ emacs Hello.tex &
```

まず、C-s を押すと dataproc1 に Hello.tex が作られます。さて、次の TeX ファイルを作成しましょう。

- ファイル名 : **Hello.tex**

```
1 \documentclass{jarticle}
2 \begin{document}
3   Hello World! こんにちは
4   \[ \int dx = x + C. \]
5 \end{document}
```

Windows 環境やエディタによっては、上のバックスラッシュ『\』は円『¥』かもしれません。TeX ファイルの拡張子は『tex』でなければなりません。

次に作成した TeX ファイルをコンパイルして dvi ファイルを作成します。端末で、ファイルのあるディレクトリから次のように入力します。

```
1 user@debian:~/Desktop/dataproc1 $ platex Hello.tex
2 ....
3 user@debian:~/Desktop/dataproc1 $
```

すると、TeX ファイルの置いてあるディレクトリに複数のファイルが生成されます。

- Hello.dvi : dvi 文章ファイル
- Hello.aux : 数式番号など、後の処理に必要な情報が書かれています。
- Hello.log : コンパイル時の処理やエラーメッセージなどが記録されます。

とりあえず、重要なのは dvi ファイルです。端末から dvi ファイルを閲覧するには次のようにします。

```
1 $ xdvi Hello.dvi
```

dvi ファイルをダブルクリックしてもよいです。dvi ファイルは直接編集することはできません。

dvi ファイルや dviout は軽量なのでコンピューターの能力が低い頃から多く使われてきましたが、最近では pdf ファイルの方が便利でしょう。PDF ファイルならばスマートフォンを含め、多くの環境で閲覧することができます。dvi ファイルを pdf ファイルに変更するには次のように dvipdfmx を使います。

```
1 $ dvipdfmx Hello.dvi
```

これで Hello.pdf という PDF ファイルが生成されます。PDF ファイルを閲覧するには

```
1 $ evince Hello.pdf
```

とします。tex ファイルから pdf ファイルを直接作るには 2 つの命令を『;』でつなげて

```
1 $ platex Hello.tex; dvipdfmx Hello.dvi
```

とやればよいです。

L^AT_EX によって文章作成をするときの注意点をまとめておきます。

1. T_EX 文章を書くときは\documentclass から始まるファイルとして入力する。
2. T_EX ファイルはコンパイルをして、dvi ファイルを作成して初めて見る事が出来る。
3. dvi ファイルには手を加えることはできない。
4. 文章を変更したいなら、元の T_EX ファイルを修正して、再度コンパイルする。
5. dvi ファイルを閲覧するには dviout(Windows) を使う。Linux なら xdvik。
6. dvi ファイルから pdf ファイルを作成するには dvipdfmx を使う。
7. Windows 等ではスラッシュ\を入力すると ¥ (小文字の円マーク) が出る。Windows ではこれらは同じですが、他の環境では異なるかもしれない。

3.5 T_EX とエラー

T_EX では命令にミスがあるとコンパイル時にエラーが出て先に進まなくなります。たとえば先ほどと同じファイルで begin の部分を bigin に変えたファイルを作ってみましょう。

- ファイル名 : **Hello2.tex**

```

1 \documentclass{jarticle}
2 \begin{document}
3   Hello World!
4   \[ \int dx = x + C. \]
5 \end{document}

```

このファイルを platex でコンパイルするとエラーが出ます。

x [Enter] を押してエラーメッセージから抜けます

エラーの内容は ファイル名.log に書かれています。

3.6 日本語を含む文章の作成についての注意

以前は、日本語を含む T_EX の文章を作成するには文字コードに注意していましたが、いまでは、文字コードは UTF-8 にして書くのが普通です。Emacs で、現在どの文字コードを使っているかは、下部のバーの左の方にあるアルファベットで表示されています。その部分が『u』= unicode, 『j』= JIS コード, 『S』=Shift-JIS, 『E』=EUC コードとなります。文字コードが混在したファイルを作ると文字化けして取り返しのつかないこととなります。

3.7 もっと長い文章の作成

もっと長い文章を作ってみましょう。

- ファイル名 : **mathproblem.tex**

```

1 \documentclass[a4j,10pt]{jarticle}
2 \begin{document}
3
4 \title{ある数学の問題と解答}
5 \author{信州太郎}
6 \date{平成24年7月20日}

```

```

7 \maketitle           % タイトルを作る
8
9 \noindent{\Large \textbf{問題}}  \\\
10 関数 $f:\mathbf{R}\rightarrow\mathbf{R}$ がすべての $x,y\in\mathbf{R}$ に対して
11 \[
12     f[x^2+f[y]] = y + [f(x)]^2
13 \]
14 を満たしているならば、 $f(x)=x$ である事を示せ。\\
15
16 \noindent{\Large \textbf{解答}}  \\\
17 上式で $x=0$ または $y=0$ とするとそれぞれ
18 \begin{equation}
19 f[f[y]] = y, \quad f[x^2] = [f(x)]^2
20 \end{equation}
21 となる。 $y$ の代わりに $f[y]$ を代入すると $f[x^2+f[f[y]]] = f[y]+[f(x)]^2$ 
22 となるが、いま得た式をつかうと、すべての $x,y$ に対して
23 \[
24     f[x^2+y]=f[x]^2+f[y]
25 \]
26 となる事が分かる。つまり $x\geq 0$ ならば $f[x+y]\geq f[y]$ となり $f[x]$ が
27 非減少関数であることが判明する。
28 もし $f[x]<x$ であるとする $x=f[f[x]]\leq f[x]$ となり矛盾。
29 また $f[x]>x$ であるとしても $x=f[f[x]]\geq f[x]$ となり矛盾。
30 したがって、 $f[x]=x$ でなければならない。
31 \end{document}

```

3.8 LaTeXの基礎・数式の入力

TeXのソースファイルは

```

1 \documentclass[a4paper,12pt]{jarticle}
2 \begin{document}
3   (本文)
4 \end{document}

```

のような形式をとります。上のjarticleの変わりに、例えば次のようなものを書きます

用途	欧文	和文	和文(新)
論文・レポート	article	jarticle	jsarticle
長い報告書	report	jreport	—
本	book	jbook	jsbook

[a4paper,12pt]は、フォントサイズ、用紙サイズを指定しています。他の選択肢は

- 10 pt (デフォルト)
- 11 pt
- 12 pt

のどれかです。用紙サイズは

- a4paper (デフォルト)
- b5paper
- b4paper
- b5paper

などがあります。`\documentclass{...}`と`\begin{document}`の間をプリアンブルといい、ここに文章の設定を書きます。本文は`\begin{document}`と`\end{document}`の間に書きます。

3.9 いくつかの注意

以下では \LaTeX の説明をするときには、断らない限り`\begin{document}`と`\end{document}`の間の本文についてだけ説明します。各自コンパイルして、 \TeX の結果を確認する場合には、`\documentclass{...}`や`\end{document}`等を忘れずに書きましょう。新しい \TeX ファイルを作成するときには、いままで自分が作った \TeX ファイルをコピーして、必要な部分を使い回すのが普通です。

3.10 特別な意味を持つ文字

次の文字

`& _ { } \ ^ ~`

は \TeX では特別な意味を持ちます。これら予約された特殊な文字は特別な意味を持ちます。たとえば

- 『\』 : 多くのコマンドは\から始まる
- 『%』 : コメントアウトするとき先頭につける
- 『\$』 : 数式環境の始まりと終わりに用いる

これらの文字を打ち込んだ通りに出力するには`\verb+...+`や`\verb_..._`を用います。たとえば`\%{}&`と書きたい場合は`\verb+\%{}$+`とします。改行も含めて、入力したままの出力をしたい場合には`\begin{verbatim}... \end{verbatim}`環境を使います。

4 文字のサイズと欧文書体

文字サイズを変えるにはたとえば次のようにします。

入力	結果
<code>{\Large 文字サイズを変える}</code>	文字サイズを変える

ここで`\Large`の部分を変えることによってサイズを変えます：

<code>\tiny</code>	5 ポイント	このぐらいの大きさ Sample
<code>\scriptsize</code>	7 ポイント	このぐらいの大きさ Sample
<code>\small</code>	9 ポイント	このぐらいの大きさ Sample
<code>\normalsize</code>	10 ポイント	このぐらいの大きさ Sample
<code>\large</code>	12 ポイント	このぐらいの大きさ Sample
<code>\Large</code>	14.4 ポイント	このぐらいの大きさ Sample
<code>\huge</code>	20.74 ポイント	このぐらいの大きさ Sample
<code>\Huge</code>	24.88 ポイント	このぐらいの大きさ Sample

4.1 数式環境

本文中に数式を入れるには、次のように $\$ \dots \$$ で囲んで書きます。

```
1 右辺の級数は $|w|^2 < \rho$ のとき絶対収束する
```

行を変えて中央に数式を書く場合は、

```
1 \[ f'(z) = \sum_{n=1}^{\infty} n a_n z^{n-2} \]
```

の様に $\[\dots \]$ で囲んで書きます。数式番号を自動で付けたい場合は、

```
1 \begin{equation}
2 N-P = \frac{1}{2\pi i} \int_C \frac{f'(z)}{f(z)} dz
3 \end{equation}
```

の様に `equation` 環境を使います。

4.2 amsmath と AMSFonts

数学でよく使用する環境・フォントに `amsmath`, `AMSFonts` があります。AMS というのは American Mathematical Society の頭文字です。`amsmath` を使えばより複雑な数式を入力する事ができるようになります。`AMSFonts` を使うと \mathbb{A} や \mathfrak{abcde} のような特殊な文字をかけるようになります。これらを使用するにはプリアンブルに`\usepackage{amsmath,amssymb}`と書いておく必要があります。つまり \TeX のソースファイルを次のように書きます：

```
1 \documentclass[a4paper,12pt]{jarticle}
2 \usepackage{amsmath,amssymb}
3 \begin{document}
4   (本文)
5 \end{document}
```

上の文字は \mathbb{A} , \mathfrak{abcde} で表示します。`amsmath` や `AMSFonts` の命令は、ここにはとても書ききれないので必要に応じて本やインターネットで調べて使いましょ。

5 今日の課題

A 君は LaTeX で次のような文章を作成したいと考えた。

<p style="text-align: center;">いろいろな数学</p> <p style="text-align: center;">信大ナナ 平成 27 年 4 月 22 日</p> <p>1 雑多なことのまとめ</p> <p>通分</p> $\frac{A}{B} + \frac{C}{D} = \frac{AD + BC}{BD}$ <p>は小学校で習う。和の記号は</p> $\sum_{n=1}^N a_n = a_1 + a_2 + a_3 + \dots + a_N \quad (1)$ <p>で定義される。無限和は</p> $\sum_{n=1}^{\infty} a_n := \lim_{N \rightarrow \infty} \sum_{n=1}^N a_n \quad (2)$ <p>と定義される。加法定理</p> $\sin(\theta + \phi) = \sin(\theta)\cos(\phi) + \cos(\theta)\sin(\phi) \quad (3)$ $\cos(\theta + \phi) = \cos(\theta)\cos(\phi) - \sin(\theta)\sin(\phi) \quad (4)$ <p>は高校で習う。部分積分の公式は</p> $\int_{\alpha}^{\beta} f(x)g'(x)dx = [f(x)g(x)]_{\alpha}^{\beta} - \int_{\alpha}^{\beta} f'(x)g(x)dx \quad (5)$ <p>である。D を C の領域とし、f(z) は D で正則であるとする。このとき D 内の任意の区分的に めらかな Jordan 閉曲線 C と C で囲まれる領域の内部の点 z_0 に対して</p> $f(z) = \frac{1}{2\pi i} \int_C \frac{f(\zeta)}{z - \zeta} d\zeta \quad (6)$ <p>が成り立つ。</p> <p>集合の演算</p> $(A \cap B)^c = A^c \cup B^c \quad (7)$ <p>が成り立つ。二つの集合 A と B が A = B であることの必要十分条件は A ⊂ B かつ B ⊂ A である。</p> <p>2 行 2 列の行列の積は</p> $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} h & i \\ j & k \end{pmatrix} = \begin{pmatrix} ah + bj & ai + bk \\ ch + dj & ci + dk \end{pmatrix} \quad (8)$	<p>と定義される。行列の積の行列式は、それぞれの行列の行列式の積に等しい：</p> $\begin{vmatrix} a & b \\ c & d \end{vmatrix} \begin{vmatrix} h & i \\ j & k \end{vmatrix} = \begin{vmatrix} ah + bj & ai + bk \\ ch + dj & ci + dk \end{vmatrix} \quad (9)$ <p>リーマンゼータ関数に対して次の公式が成り立つ</p> $\zeta(s) = \prod_{p \text{ prime}} \frac{1}{1 - p^{-s}}. \quad (10)$ <p>曲線 $\gamma: [0, 1] \rightarrow \mathbb{R}^2$ は連続微分可能であるとする。このとき曲線 γ の長さ L は</p> $L = \int_0^1 \sqrt{x'(t)^2 + y'(t)^2} dt \quad (11)$ <p>で与えられる。ただし $\gamma(t) = (x(t), y(t))$。</p>
---	--

A 君は一通り入力したものの、多くのタイプミスをしてしまい、TeX ファイルをコンパイルすることができなかった。A 君のファイルは次のアドレスからダウンロードすることができる。

<http://math.shinshu-u.ac.jp/~isasaki/classes/2019dp1/files/02report.tex>

この A 君のファイルを修正し、A 君が目標とする結果になるようにせよ。ただし、「信大ナナ」の部分を自分の学籍番号と名前にすること。

(ヒント) エラー箇所はたくさんあり、どこが間違っているのかを見つけるのは難しい。そこで次のようにするとよい

1. `begin{document}` と `end{document}` の間の行の先頭に `%` を入れてコメントアウトする。
2. `platex` でコンパイルし、正常にコンパイルされることを確かめる。
3. 先頭から少しずつコメントを外しながらエラーを修正していく。

PDF ファイルを作ることができたら eALPS から提出すること。提出期限は eALPS の課題の場所にある。