

第X回 情報工学演習B(夜間主コース) レポート

123456789-0 ○○○○

室蘭 情報

提出日：20xx 年 x 月 x 日

1 はじめに

本レポートでは、 \TeX を使用したレポート作成方法の概略について説明する。使用しているスタイルファイル (jikken_report) は実験レポートのためのスタイルオプションファイルである。

本レポートのようにスタイルファイルを使用する場合、参照する \TeX ファイルと同じディレクトリもしくは \TeX コンパイラの参照パスが通っている場所 (`c:/tex/share/texmf/ptex/platex/base/`, デフォルトでインストールした場合のフォルダ) に使用するスタイルファイルを置いておく必要がある。

ここでは \TeX に関する詳細 (コマンド, 環境など) については、一切ふれない。 \TeX に関する詳細は、奥村さんの書籍^{?)}, \TeX Wiki^{?)}, 乙部さんの書籍^{?)}などを参考に各自勉強すること。また、Windowsでの \TeX インストールに関しては、<http://is.csse.muroran-it.ac.jp/~sin/lecture/>^{?)}を参照すること。

以下、下記の事柄について説明する。

- 演習レポートの書き方
- プログラムソースの載せ方
- プログラム実行例の載せ方
- 図表の書き方, 参照方法
- 式の書き方, 参照方法

2 演習レポートの書き方

演習レポートは、基本的に以下の構成からなる。各部分が各章になるが、章のタイトルは内容に合わせて変更することとなる。

- はじめに
- 作成したプログラム
- プログラムの実行例
- 工夫した点
- 結果の考察・まとめ

「はじめに」の章では提出するレポートの目的・作ろうとしているものの説明を記述する。

「作成したプログラム」の章では自身が作成したプログラムのソースを載せる。プログラムのソースを載せる場合には工夫が必要であるので、第??章を参考にするとよい。

「プログラムの実行例」の章では、作成したプログラムのコンパイルのコマンド、実行のコマンド、実行したコマンドに対する入力 (入力例), 入力に対する実行結果を載せる。第??章および第??章を参考にするとよい。

「工夫した点」の章では、プログラムを作成した時に自身で工夫した点があれば、どのように考え、どのような工夫をしたか、記述してもらえると嬉しい。

「結果の考察・まとめ」の章では、できた結果をもとに論理的に読み取れる内容を説明すること。各結果の考察に加えて全体的な説明をまとめとして記述すること。

3 プログラムソースの貼り方

プログラムソース中には $\text{T}_{\text{E}}\text{X}$ における特殊記号が多数含まれるため、一般には`verbatim`環境を利用する。一例を下記に示す。

```
Hello world —
/*****
名前:  hello.c
機能:  ウェルカムメッセージを表示する
入力:  なし
出力:  なし
作者:  (自分の学籍番号と名前)
制作日:  2008/10/15
特記事項:  特になし
*****/

#include <stdio.h>

int main( void )
{
    printf("ようこそ C 言語の世界へ!");
    printf("ようこそ C 言語の世界へ!\n");
    printf("ようこそ\n C 言語の世界へ!\n");

    return 0;
}
```

4 プログラムの実行例の載せ方

プログラムの実行に関しても、ソースの場合と同様 $\text{T}_{\text{E}}\text{X}$ における特殊記号の影響を避けるため `verbatim` 環境を利用する。下記に一例を示す。

```
コンパイルと実行 —
$ gcc -o hello hello.c
$ ./hello
ようこそ C 言語の世界へ!ようこそ C 言語の世界へ!
ようこそ
C 言語の世界へ!
```

5 プログラムの実行結果の保存方法

レポートに実行結果を貼り付ける場合、作成したプログラムを `x11(kterm)` 上で実行し、その結果をコピーして `emacs` に貼り付ける必要がでてくる。しかし現在、コンピュータの問題で、`x11(kterm)` 上の日本語を `emacs` にコピーして張り付けることが出来ない。

ここでは、レポートに実行結果を貼り付ける 2 つの方法を提示しておく。やりやすい方法で実行結果をレポートに貼り付けること。

5.1 script

script というコマンドで, x11(kterm) 上の表示されている文字情報を全て記録することができる. x11(kterm) 上で

```
script
```

とコマンドを打つと, それ以後の表示が全て記録される. これを終了するには

C-d (Control キーを押しながら d)

を入力する. すると script というコマンドを入力したところから C-d を押したところまでが "typescript" というファイルに全て記録される. script 作成した実行ファイル C-d

とすることで typescript に実行結果を保存することができる. 後は, typescript を emacs で読み込み, コピーし, レポートの文書に貼り付ける.

script でコンソールの情報を取得すると, ^M という文字が出てくる. これを自動的に消す方法. 以下のコマンドを実行. `nkf -w -Lu -overwrite typescript`

5.2 emacs で shell

emacs にはシェル機能があり, emacs 上で x11(kterm 等) を実行することができる. やり方は, M-x shell (M-x はアップルキーを押しながら x)

とすることで emacs のウインドウが分割され, x11(kterm) と同じようなコマンド入力待ちになる. ここで作成したプログラム等を実行した結果をコピーし, レポート文書に貼り付ける.

6 図表の書き方

6.1 図について

図を出力するためには, 下記のような書式で記述する.

```
\begin{figure}[htbp]
\begin{center}
\includegraphics[width=0.7\linewidth]{eps/graphic_data.eps}
\end{center}
\caption{図形}
\label{fig::graphic}
\end{figure}
```

`\begin{figure}[htbp]` における [htbp] は図を文章のなかに入れ込む位置に関するパラメータであり, Here(この場所で), Top(ページの上で), Bottom(ページの下で), Page(次のページにおいて) の頭文字を意味する. 論文などにおいて, 図はページの上もしくは下に挿入することが一般的であるためオプションとしては, `\begin{figure}[tbp]` を勧める.

また, `\label{ラベル名}` はこの図形を自動参照するためのラベル付けである. 図を参照する場合には, このラベルを参照して `\fgref{ラベル名}` と記述する¹. この場合の例では, `\fgref{fig::graphic}` と記述することにより, 「図 ??」となる.

なお, 図におけるキャプションは**必ず図の下**につけること.

¹fgref は, Figure(図形) の Reference (参照) を意味する.

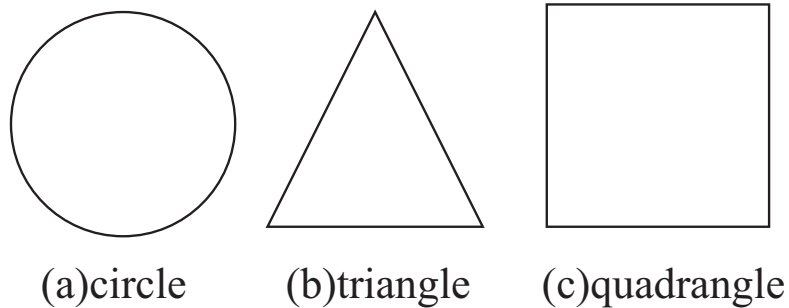


図 1: 図形

表 1: Problem Instance.

Problem	N	W	\bar{w}	$\sigma(w)$
tai75a	75	1445	183.4	242.9
tai75b	75	1679	198.7	273.4
tai100a	100	1409	152.0	201.5
tai150a	150	1544	145.54	200.7

6.2 表について

表を出力するためには，下記のような書式で記述する．

```
\begin{table}[tbp]
\begin{center}
\caption{Problem Instance.}
\label{tb::example}
\begin{tabular}{ccccc}
\hline
Problem &  $N$  &  $W$  &  $\bar{w}$  &  $\sigma(w)$  \\
\hline \hline
tai75a & 75 & 1445 & 183.4 & 242.9 \\
tai75b & 75 & 1679 & 198.7 & 273.4 \\
tai100a & 100 & 1409 & 152.0 & 201.5 \\
tai150a & 150 & 1544 & 145.54 & 200.7 \\
\hline
\end{tabular}
\end{center}
\end{table}
```

基本的には，図における場合と同様に`\begin{table}[htbp]`により表の文中での位置を決め，`\label{ラベル名}`で表のラベル付けを行う．また，表を参照する際には`\tbref{ラベル名}`と記述

する²。この場合の例では、`\tbref{tb::example}`と記述することにより、「表 ??」となる。
 なお、表におけるキャプションは**必ず表の上**につけること。

7 式について

式の書き方には幾つかの種類がある。以下、それぞれについて示す。

1. 文中で式を入れ込む場合

`$数式$`という書き方をする。例えば、`$\cfrac{b_{2}}{a^2}\leq 0$`と記述すると、「 $\frac{b_2}{a^2} \leq 0$ 」のように出力される。

2. 数式を番号なしで 1 行挿入する場合

`\[\]` と記述する。

例えば、`\[\cfrac{b_{2}}{a^2}\leq 0 \]` と記述すると、下記のように出力される。

$$\frac{b_2}{a^2} \leq 0$$

3. 数式を番号つきで 1 行挿入する場合

`\begin{equation} 式 \end{equation}`と記述する。

例えば、`\begin{equation} \cfrac{b_{2}}{a^2}\leq 0 \end{equation}`と記述すると、下記のように出力される。

$$\frac{b_2}{a^2} \leq 0 \tag{1}$$

4. 数式を番号なしで複数行挿入する場合

`\begin{eqnarray*} 式 1 \\ 式 2 \end{eqnarray*}`と記述する。

例えば、`\begin{eqnarray*} \cfrac{b_{2}}{a^2}\leq 0 \\ a\times b \geq 10 \end{eqnarray*}`と記述すると、下記のように出力される。

$$\begin{aligned} \frac{b_2}{a^2} &\leq 0 \\ a \times b &\geq 10 \end{aligned}$$

5. 数式を番号つきで複数行挿入する場合

`\begin{eqnarray} 式 1 \\ 式 2 \end{eqnarray}`と記述する。

例えば、`\begin{eqnarray} \cfrac{b_{2}}{a^2}\leq 0 \\ a\times b \geq 10 \end{eqnarray}`と記述すると、下記のように出力される。

$$\begin{aligned} \frac{b_2}{a^2} &\leq 0 & (2) \\ a \times b &\geq 10 & (3) \end{aligned}$$

なお、番号付きの式には図、表と同様にラベル付けおよびその参照を行うことができる。一例を以下に示す。

²`\tbref` は、Table(表) の Reference (参照) を意味する。

```

\begin{eqnarray}
\cfrac{b_{2}}{a^2}\leq 0 \quad \text{\label{eq::const-1}} \\
a\times b \geq 10 \quad \text{\label{eq::const-2}}
\end{eqnarray}

```

$$\frac{b_2}{a^2} \leq 0 \tag{4}$$

$$a \times b \geq 10 \tag{5}$$

式を参照するときには、`\eqref{式ラベル}`と記述する．上記の例では`\eqref{eq::const-1}`と記述することにより「式(??)」が出力される．

ちなみに図，表，式だけでなく章 (section), 節 (subsection) など様々な箇所でもラベル付けおよびその参照を行うことができる．その場合には、`\ref{ラベル}`と記述することによってラベル付けされた場所の番号を呼び出すことができる．そのため，たとえば章の場合、`\ref{ラベル}章`といった書き方を行うことになる．