

第1問 (10点)

標準的なタッチタイピングでは, 以下の10個のキーは左右どちらの手のどの指を使って入力するのか答えなさい.

[c] [e] [l] [n] [p] [t] [u] [v] [w] [z]

解答例

[b] 左人差し指

第2問 (10点)

銀行のATMの暗証番号は4桁の数字でおこなうが, 多くのコンピュータシステムではパスワードは6桁以上の英語, 数字, 記号を使うことが多い. なぜ, 経済的な価値がより高いと思われる銀行の方がより安全性の低い暗証番号を採用しているのか. 考えられる理由を箇条書きにして2つ以上提示しなさい.

第3問 (15点)

以下の用語に関して50~100文字程度で解説を書きなさい.

- WWWサーバ
- WWWブラウザ
- ロボット型検索エンジン

第4問 (20点)

以下に示すHTMLファイルがある(行頭の3文字は行番号で実際のファイルには含まれないものとする).

```
1: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2: <html lang="ja">
3: <head>
4: <title>self introduction of Todai Taro</title>
5: </head>
6: <body>
7: <h1>東大太郎のホームページ</h1>
8:
9: <p>
10: 私の名前は東大太郎です。
11: 現在<a href="http://www.u-tokyo.ac.jp/">東京大学</a>に通っています。
12: </p>
13:
14: <p>
15:
16: いろんな趣味があります。
17: </p>
18: <ul>
19: <li>音楽鑑賞
20: <li>映画鑑賞
21: <li>絵画鑑賞
22: <li>読書
```

24:  
25: <p>  
26: みなさん宜しくお願ひします。  
27: </p>  
28: </body>  
29: </html>

上の HTML ファイルの中で、以下に該当する行番号をすべてあげなさい。該当する行番号がない場合は「なし」と書いてください。

- (a) 文書外部へのリンクのアンカー (anchor) 要素を含む行
- (b) 順序つきリスト (ordered list) の終了タグを含む行
- (c) 見出し (Heading) 要素の開始タグを含む行
- (d) 順序なしリスト (unordered list) の項目要素を含む行
- (e) HTML 要素の終了タグを含む行
- (f) 文書型宣言を含む行
- (g) 文書のタイトル (title) を表す要素の開始タグを含む行
- (h) 文書内部へのリンクのアンカー (anchor) 要素を含む行
- (i) 段落 (paragraph) 要素の開始タグを含む行
- (j) 埋め込み画像 (image) 要素を含む行

#### 第 5 問 (10 点)

以下の文章の空欄を埋めなさい。

情報処理の文書清書システムの課題をやるために、Emacs を使って、~/jousho03 の下に kadai.tex という名前で以下のようなファイルを作った。

```
\documentstyle[graphics]{jarticle}
\begin{document}
\title{遠足のお知らせ}
\author{340395 田中哲朗}
\date{2003 年 7 月 30 日}
\maketitle
```

本年も以下の要領で遠足を実行しますので、みなさまお誘ひ合わせの上、ぜひ御参加下さい。

```
\section{日程表}
\begin{tabular}{|l|l|l|} \hline
日 & 時間 & 予定 \\ \hline
7 月 77 日 & 8:00 & 情報教育南棟前集合 \\ \hline
同日 & 12:00 & 昼食 (噴水前) \\ \hline
同日 & 17:30 & 情報教育北棟前解散 \\ \hline
\end{tabular}
\sectoin{行き先}
目黒区駒場 3-8-1
```

```

\begin{itemize}
\item 弁当, 水筒, おやつ
\begin{enumerate}
\item おやつは{\bf 300}円以内.
\item Bananas は{\it おやつ}に含めない.
\item {\large Juice}は不可.
\end{itemize}
\item 画用紙, クレヨン
\item 傘{\footnotesize (雨が降りそうな場合)}
\item 必要に応じて以下の式も持参すること.

$$H(X) = \sum_{i=1}^n (-p_i \log p_i)$$

\end{itemize}
\section{注意}
途中で以下のようなかぼちゃを見かけたら報告するように.
\begin{center}
\includegraphics{pumpkin.eps}
\end{center}
\end{document}

```

kterm のウィンドウで

```
platex kadai.tex
```

を実行したら,

```

This is JTeX, Version 1.8, based on TeX Version 3.14159 (Web2C 7.2)
! I can't find file 'kadai.tex'.
<*> kadai.tex

```

Please type another input file name:

と出て止まってしまったので, [ctrl]+[c] を押して終了した. kterm のウィンドウではカレントディレクトリが「~」になっていることに気がついたので, kterm のウィンドウのウィンドウで,

(a)

と入力して, カレントディレクトリを「~/jousho03」にしてから, 再び

```
platex kadai.tex
```

を実行したら,

```

! Undefined control sequence.
1.18 \sectoin
      {行き先}

```

というエラーが出た.

(b)

の行を

(c)

! LaTeX Error: \begin{enumerate} on input line 25 ended by \end{itemize}.

See the LaTeX manual or LaTeX Companion for explanation.

Type H <return> for immediate help.

...

1.27 \end{itemize}

というエラーが出た .

(d)

の行を

(e)

としたところ、無事に platex コマンドが終了して kadai.dvi というファイルができた . そので、

xdvi kadai.dvi

を実行して、画面で確認してみたところ、 $p_i$  とするつもりが、 $p^i$  になっていたことに気がついたため、

(f)

の行を

(g)

と直して、レポートを提出した .

第 6 問 (10 点)

以下に、画像形式の 1 つ PPM 形式のファイルのサンプルと、その説明をつける .

```
P3
4 4
255
 0 0 0 0 0 0
 0 0 0 255 0 255
 0 0 0 0 0 0
255 0 255 255 0 255
 0 0 0 255 0 255
255 0 255 255 0 255
255 0 255 255 0 255
255 0 255 255 0 255
```

P3 の部分がマジックナンバーという部分で、このファイルが他の画像ファイルではなく PPM 形式であることを示す (Windows の多くのアプリケーションのように、拡張子が .ppm であることで判断しているわけではない) . PPM 形式では空白 (White Space, Space コードだけではなく改行、タブも含まれる) はいくついても良いが、P3 の前に入れてはいけない . マジックナンバーの後に 空白があり、

```
4 4 255
```

それぞれ 0(暗い) から 255(明るい) までの 256 階調の色は人間に自然な画像を表せるといわれているので、ここでは 255 を指定している。

残りの部分で、 $4 \times 4 = 16$  ピクセルの RGB 値を順に指定している。ピクセルの指定の順番は横書きの文章と同じように左上から右下へ横方向に走査していく。1 ピクセル内の指定の順番は R G B の順番になっている。

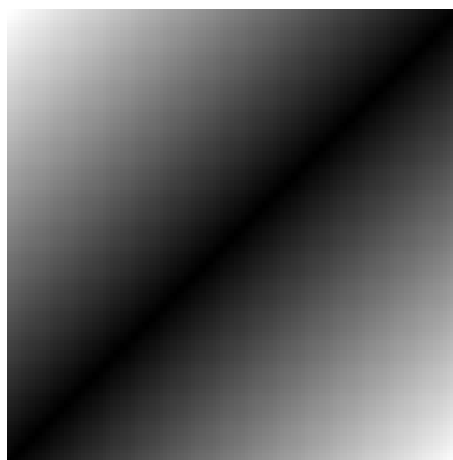
この画像ファイルはどのような図形を表しているか、図形と色を以下のような形式で書きなさい。



### 第 7 問 (25 点)

以下の文章の空欄 を埋めなさい。

Java 言語で、以下のような画像を作る PPM ファイルを作成するプログラムを作ろうとした。



以下のように Grad.java というファイルを作り、

```
class Grad{
    public static void main(String[] args){
        int width=128, height=128;
        System.out.println("P3 "+width+" "+height+" 255");
        int i,j;
        for(i=0;i<height;i=i+1){
            for(j=0;j<width;j=j+1){
                // Math.abs は絶対値の計算
                int k=Math.abs(i-j)*2;
                System.out.println(k+k+k);
            }
        }
    }
}
```

(a)

と kterm 上で入力してコンパイルして、

(b)

と実行してみて、grad.ppm というファイルを作成してみたが、

```
convert grad.ppm grad.gif
```

を実行すると、

```
convert: not enough pixels (grad.ppm).
```

のようなエラーを出して終了してしまった。grad.ppm を Emacs で内容を見てみると、

```
P3 128 128 255
```

```
762
```

```
756
```

```
750
```

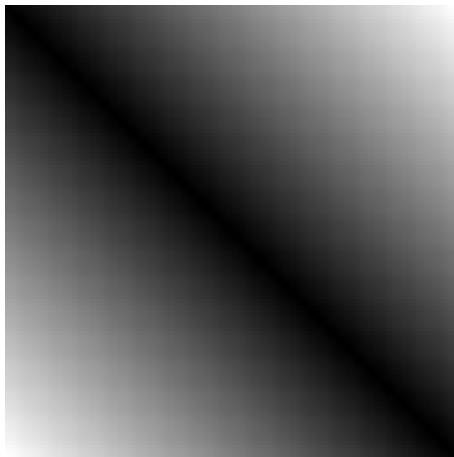
のように、R G B 値が並ぶのではなく、全部が足された値が並んでいる。そこで、元のプログラムの

(c)

の行を

(d)

と修正したところ、今度は convert まで済んだが、変換してできた grad.gif ファイルを Mozilla で表示させたところ、



のような図が出てしまった。座標値から明るさを求める式が間違っていたことに気がついたので、プログラムの

(e)

の行を

(f)

と修正したところ、想定する出力が得られた。