

TR-A-0002

文字認識機能を持つ
文字フォントエディター

F a b l e

F a b l e : A FONT EDITOR WITH CHARACTER
RECOGNITION FUNCTIONS

森 吉弘 横澤 一彦 梅田 三千雄

Yoshihiro Mori Kazuhiko Yokosawa Michio Umeda

1987.5.25

A T R 視聴覚機構研究所

目次

| | |
|--------------------------|----|
| 1. はじめに | 1 |
| 2. フェーブルの概要 | 2 |
| 3. フェーブルの動作機能 | 3 |
| 3.1 フォントの作成方法 | 3 |
| 3.1.1 マウス入力 | 3 |
| 3.1.2 かな漢字変換（一太郎）を利用した入力 | 3 |
| 3.1.3 イメージスキャナー入力 | 4 |
| 3.2 字形評価 | 6 |
| 3.3 文字認識 | 7 |
| 3.3.1 特徴抽出 | 7 |
| 3.3.2 認識尺度 | 9 |
| 3.4 フォントの変形 | 11 |
| 3.5 作成フォントの入出力 | 12 |
| 3.6 その他の機能 | 12 |
| 4. フェーブルの使用例 | 13 |
| 4.1 文字認識法の比較 | 13 |
| 4.2 変形漢字の認識 | 13 |
| 5. むすび | 17 |

謝辞

参考文献

付録

| | | | |
|---------------|----|---|---------------------------------------|
| FABLE[feible] | n. | 1 | 寓話（ぐうわ）《動物などを擬人化したもの》 |
| | | 2 | 伝説、神話 |
| | | 3 | 作り話、作り事 |
| | | 4 | うそ、むだ話 |
| | | 5 | 早くて素晴らしい文字編集道具 |
| | | | Fast And Brilliant Letter_font Editor |

1. はじめに

このフォントエディター「*Fontable*」は簡便な文字作成と文字認識による作成文字の字形評価を目的としてPC-9801VM上に作成したものです。入力手段として、

① マウスからの入力

② イメージスキャナー(PC-IN503)からの画像ファイル

③ 一太郎(かな漢字変換)を利用したJISフォントの読み込みの3通りをサポートしており大変手軽に文字フォントが作成できます。そして、入力フォントの修正はマウスを使用することにより行えます。また文字変形機能により任意の2文字間の中間パターンを作成可能です。作成した文字フォントは外部ファイルとして保存し、ワープロの外字フォントとして利用できます。

また、このプログラム内部から動作可能な認識特徴として

① 文字線の大まかな 外部形状に着目した パリフェラル特徴

② 文字線の大まかな 複雑さに着目した 線密度特徴

③ 文字線の大まかな 濃度分布に着目した メッシュ特徴

があります。

これらの認識特徴を基に、L1距離・L2距離・類似度のいずれかを尺度として選択し、文字認識を行うことが可能です。文字作成機能と文字認識機能を密に結合することにより文字認識手法の特徴(短所・長所)や作成した文字の形状を簡単に分析評価する事ができます。本エディターの具体的な起動条件と方法は付録に示します。

2. フォントエディター（フェーブル）の概要

フォントエディターは図1に示す階層構造メニューからマウスにより種々の機能を選んで実行させる構成になっています。フェーブルは起動時に<メニューモード>となり、編集文字の移動、交換、および他のモードへの指定を行います。また、<修正モード>は、マウスの機能の選択を行います。<外部入力モード>は編集文字の入力方法の指定、<後処理モード>は編集し終った文字の登録・印刷・認識などの処理の指定をおこないます。<文字認識モード>では、編集文字に対して複数の認識手法による認識実験を行うことができます。

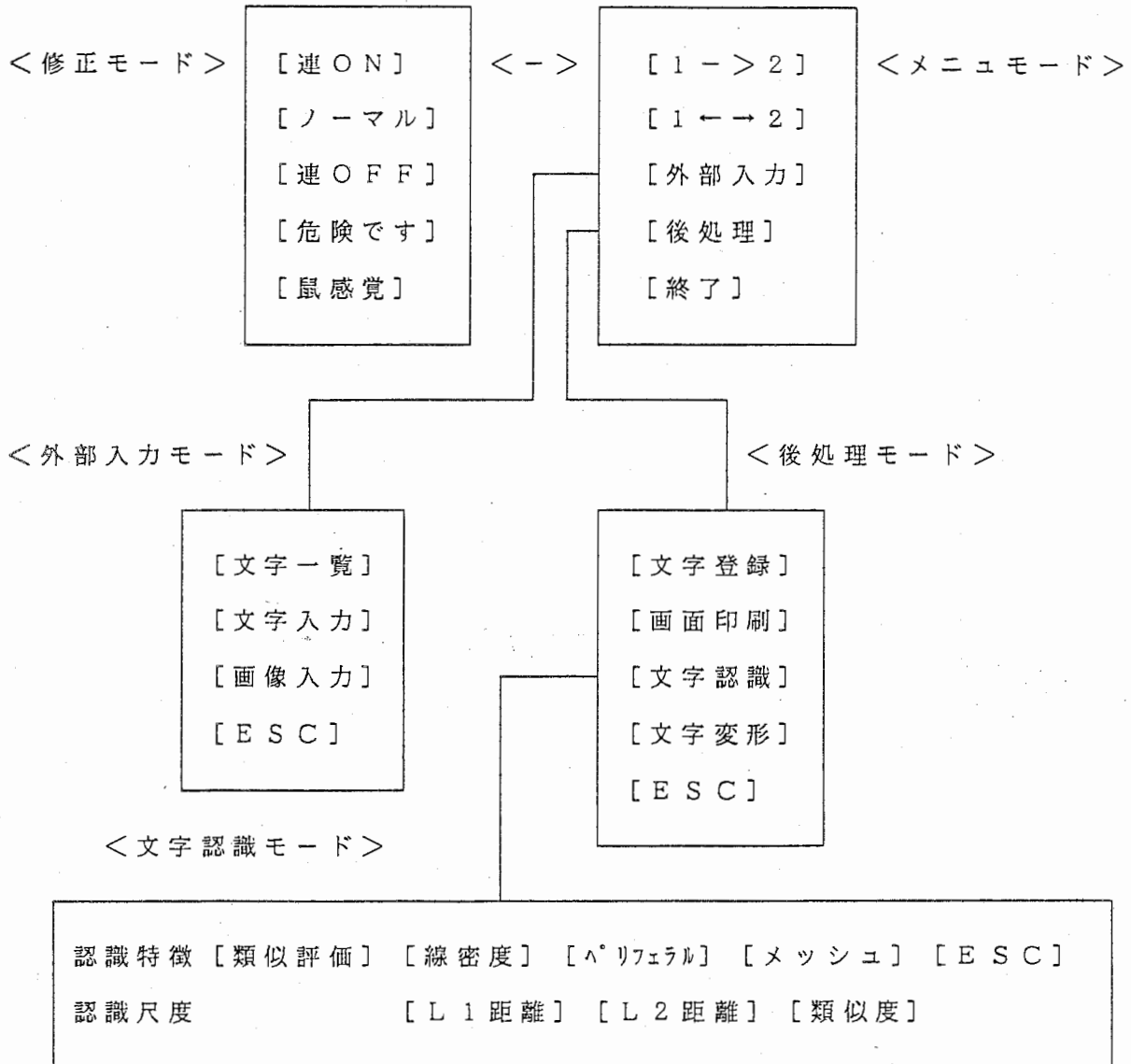


図 1 階層構造メニュー一覧

3 フェーブルの動作機能

3. 1. フォントの作成方法

3. 1. 1 マウス入力

まず、フェーブルは起動時に、かな漢字変換時に読み込むJISフォントファイル及び既作成フォントの登録ファイルの保存ディスク名を指定した後自動的にフォント作成修正モードになります。このモードでは、マウスを使用することにより24×24ドットのフォントが作成できます。キーボードからの入力は必要としません。図2がマウス入力時の画面であり、左側の編集文字用24×24ドットのマス目に作成します。なお、カーソルは編集文字と参照文字の同じ場所に表示されています。

修正モードと

して、[連続

ON][連続

OFF][ノ

ーマル]の3

種類が選択で

きます。[ノ

ーマル]モー

ドではマウ

簡易フォントエディター
左ボタン: ON OFF
連続ON ノーマル 連OFF 危険です 鼠感覚
[モード]
左ボタンで選択

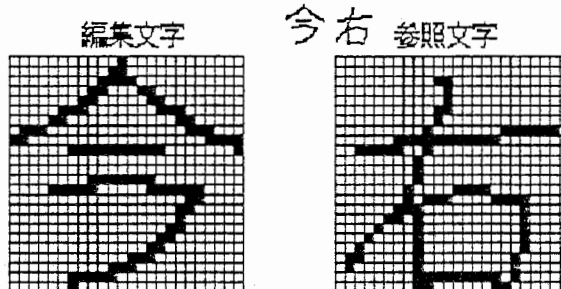


図2 マウスによる入力画面

スの左ボタンでカーソルの存在するマス目の白/黒を反転できます。また、[連続ON]ではマウスの左ボタンを押しながらマウスを動かすことにより連続してドットをONにできます。[連続OFF]は、ドットを連続して消すことができます。[鼠感覚]を選択すればマウスの感度を変更できます。修正モードにはフォント作成マス目の左上隅から上方向にマウスを動かせばいつでも入ることができます。

3. 1. 2 かな漢字変換入力

フォントを作成するために既存のフォントを参照しながら、あるいは既存のフォントをもとに修正して作成する場合に、既存のフォントとしてJIS6234のすべてのフォント(24×24ドット)を利用する

ことができます。しかしながら、このフォントを呼び出すためにJISコードや区点コードを与えることはフォント作成者にとってかなりの負担となります。そこで、本エディターでは、かな漢字変換を利用して目的とする漢字を呼び出しています。また逆に目的とする漢字のJISコード・区点コードなどを漢字を与えるだけで知る事もでき、漢字コード表としても活用可能です。この機能は<メニューモード>の「外部入力」モードで「文字入力」を選択することによって利用することができます。図3に、かな漢字変換を利用したフォント入力画面の例を示します。尚、かな漢字変換にはATOK5を使用していますので、日本語ワープロ「一太郎」と同様の手順で行うことができます。

マウスの左のボタンで文字を選ぶ

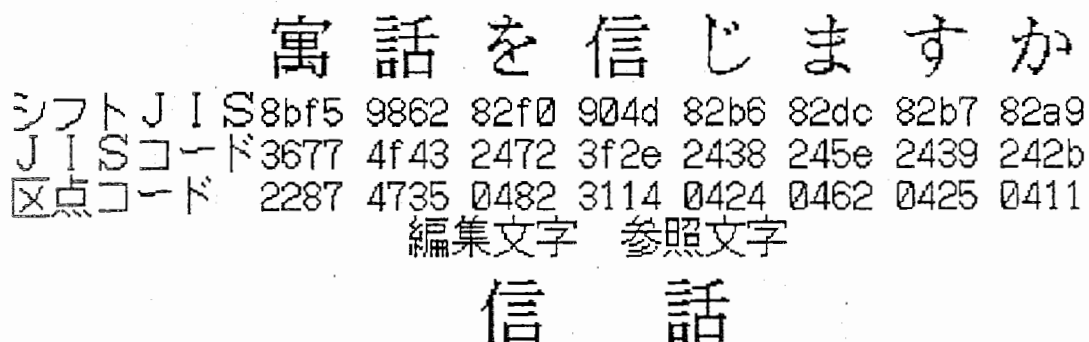


図3 かな漢字変換を利用したフォントの入力

3. 1. 3 イメージスキャナー入力

ロゴやキャラクターのフォントを作成するために、他の印刷物を参考にして作成する場合とか、文字認識を行う文字を印刷物もしくは手書き文書から取りたいときにこのイメージスキャナー入力を使用します。

イメージスキャナーから直接に読み込むことは本プログラムではサポートしていませんが、ユーティリティーとして作成しました画像読み取り

プログラム**「yomuzo」で取り込んだ画像ファイルを入力ファイルとして利用可能です。この機能は、「外部入力」モードで「画像入力」を選択することによって利用することができます。画像ファイルの取り込み例を図4に示します。取り込み画像の読み取り線密度はユーティリティー・プログラムで変更できます(3.5~9.45ドット/mm)。

- ・ 接続可能イメージスキャナー PC-IN503, 502, 501
- ** 画像読み取りプログラム [yomuzo]
 画像転送 9600ボー(RS-232C)
 読み取り速度 約3分(10×5cm)

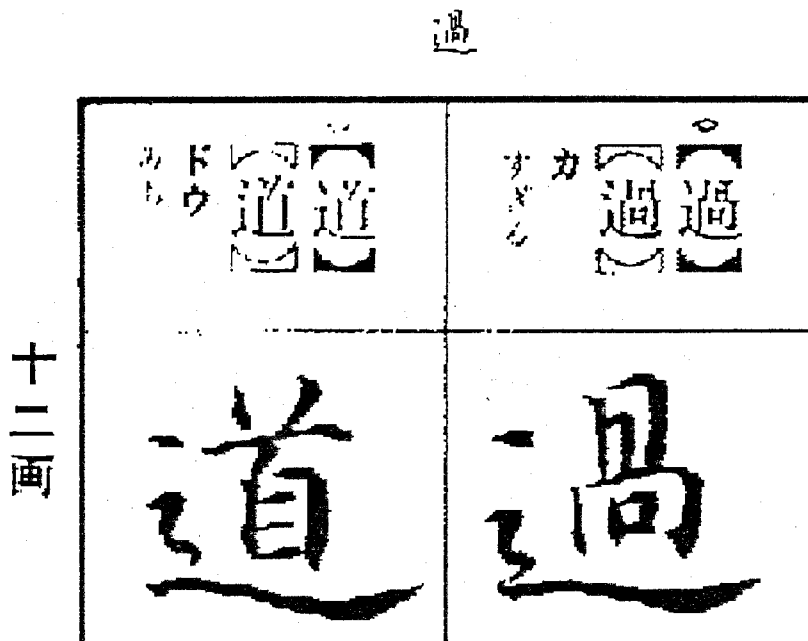


図4 画像データの取り込み例

ディスプレイ上に表示された画像データをフォントとして取り込むには、マウスを操作してフォントとして切り取るべき矩形領域を指定します。図4から切り取った例として、図4の”道”から切り取った”首”のフォントを図5に示します。

イター
N OFF
印刷 文字入力 画像入力 ESC

賀首 参照文字

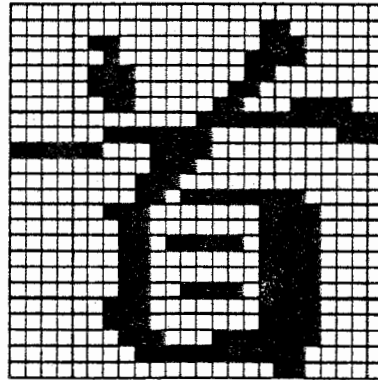


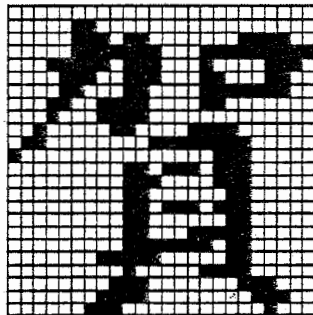
図5 画像データからのフォント作成例

3. 2 字形評価

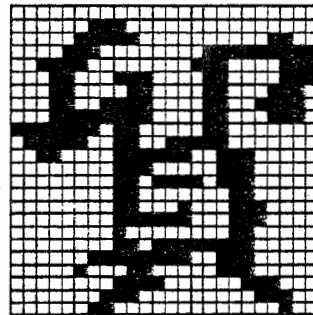
作成したフォントの字形を評価するために、「文字認識」モードで「類似評価」機能を選択することにより、編集文字と参照文字間の類似度や重なりドット数などを計算できます。尚、デ

編

編集文字



賀賀 参照文字



| | | |
|-----------|-------|---------------------|
| 左パターン | ドット数: | 169 |
| 右パターン | ドット数: | 190 |
| 重複 | | 121 |
| 単純類似度 | | 0.675 |
| ペリフェラル類似度 | 1次: | 0.9461 + 2次 : 0.933 |

図6 文字評価例

スプレイ上では文字間の重なり具合が色分けして表示されます。

3. 3 文字認識

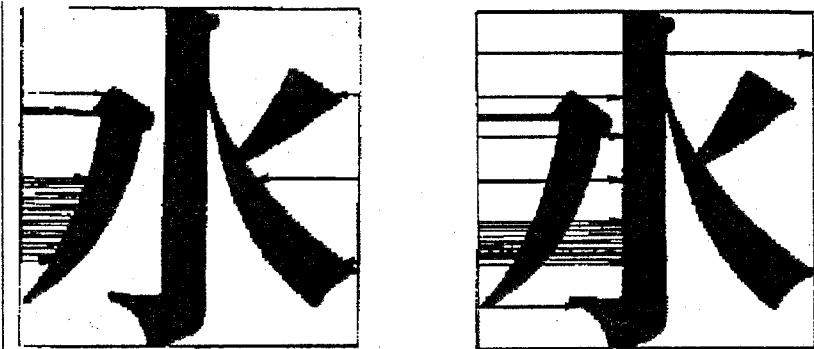
作成したフォントに対して、フェーブルから文字認識を行うことができます。この機能は「後処理」モードで「文字認識」モードを選択することにより実行することができます。文字作成機能と文字認識機能を同一プログラム内で実現したことにより、他の文字フォントと比較して、視認性の高い文字フォントの作成が可能となりました。また、文字認識手法の性能及び特徴を簡単に評価することもできます。フェーブルが標準サポートしている文字認識手法は、①ペリフェラル特徴 ②線密度特徴 ③メッシュ特徴を利用した3種類の手法です。

それぞれ手法は認識尺度としてL1距離・L2距離・類似度の3種類の尺度を選んで文字認識を行うことができます。認識アルゴリズムおよび認識尺度について示します。

3. 3. 1 特徴抽出

① ペリフェラル特徴を利用した文字認識手法

この手法は、文字パターンはストロークの輪郭形状によっても特徴付けられていると考え、周辺から見た大まかな輪郭形状



この特徴をとらえよ 図7 ペリフェラル特徴(1次:2次)

とするものです。ペリフェラル特徴は、図7に示すように24×24ドットの文字パターンの領域の外枠を大まかに分割し(フェーブルでは6分割)、分割された各部分から反対側に向かって文字部に出会うまで領域の面積を特徴として抽出します。1次特徴は最初に文字部に出会うまでの領域、2次特徴は2度目に文字部に出会うまでの領域となります。フェーブルでは、上下左右方向にそれぞれ24本走査し走査線4本分を1つの特徴として6次元ずつを、1次ペリフェラル特徴と2次ペリフェラル特徴で合計48次元の特徴を抽出し、文字認識を行っています。

② 線密度特徴を利用した文字認識手法

線密度特徴は図8に示す様に、垂直・+45度・水平・-45度方向に文字パターンを走査し、そのときに横切った文字線数を計数する事によって抽出されます。フェーブルでは、文字が24×24ドットで構成されているので垂直・水平方向は24本、±45度方向は47本走査することになります。抽出された垂

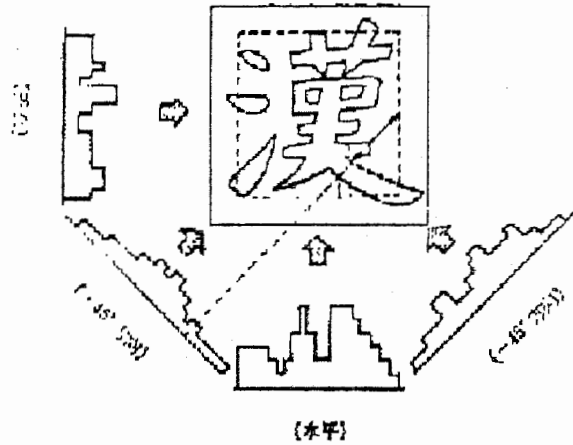


図8 線密度特徴

直方向及び水平方向の特徴は走査2本分を1つの特徴として12次元ずつ24次元の特徴となり、±45度方向の特徴は4本分を1つの特徴として12次元ずつ24次元の特徴となります(尚、12本目は重複計算します)。合計48次元の特徴を文字毎に計算し文字認識を行っています。

③ メッシュ特徴を利用した文字認識

メッシュ特徴は、文字パターンをN×Nの粗いメッシュ領域に分割し、その各々のメッシュ領域に含まれる文字部(黒点)の数を計数することにより抽出されます。フェーブルでは6×6の粗いメッシュ領域に文字を領域分割して合計36次元の特徴を文字毎に計算し文字認識を行っています。

3. 3. 2 認識尺度

L1 距離・L2 距離・類似度の認識尺度に付いて計算式を式(1)

(2) (3) に示します。

$$(L1) = \sum_{k=1}^{48} |P_k - R_k| \quad \dots \dots \dots (1)$$

$$(L2) = \sqrt{\sum_{k=1}^{48} (P_k - R_k)^2} \quad \dots \dots \dots (2)$$

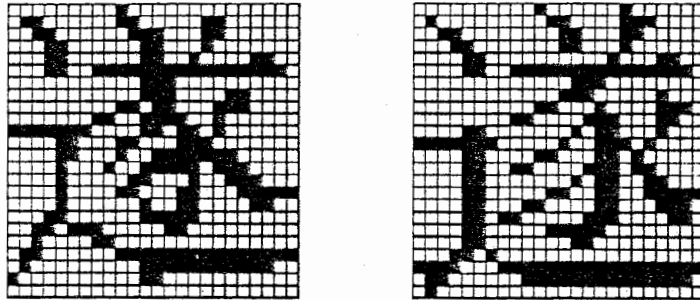
$$(\text{類似度}) = \frac{\sum_{k=1}^{48} P_k \cdot R_k}{\sqrt{\sum_{k=1}^{48} P_k \cdot P_k} \cdot \sqrt{\sum_{k=1}^{48} R_k \cdot R_k}} \quad (3)$$

(P_k は入力文字の特徴ベクトル、R_k は辞書文字の特徴ベクトル)

プログラムを変更することにより他の文字認識手法を組み込むことが可能です。ただし文字認識を実行するのがPC-9801 (CPU8086程度)なのでリアルタイムで文字認識を行うわけにはいきません。PC-9801 V X (80286+80287)を使用した場合、認識尺度に[L1]・[L2]を選択すれば約3秒、[類似度]を選択すれば約13秒程度、1文字を認識するのにかかります。認識時間のかなりかかる文字認識手法の評価をサポートするためにRS-232Cを介して上位のホスト・マシン(VAX等)と通信を行えるようにする予定です。

図9-1、9-2、9-3に、手書き文字データをイメージスキャナで読み込みフォントとして切り出した後、文字認識を各々の特徴で行った例を示します。認識尺度は何れもL1距離を用いました。図に示す様に、分類基準を満足する候補文字が最大10個表示されます。

フォントエディター Fable
 左ボタン: ON OFF
 類似評価 線密度 パリフェラル メッシュ ESC
 L1距離 L2距離 類似度
 [文字認識] 認識尺度
 編集文字 遂 遂 参照文字



私は遂を逆と読みます 左ボタン74

⁷⁴逆 ⁷⁴遂 ⁸²送 ⁸⁶超 ⁸⁷起 ⁸⁷遂 ⁸⁸逸 ⁹⁰速 ⁹¹途 ⁹¹迹

図9-1 線密度特徴認識結果 (尺度L1)

私は遂を遂と読みます 左ボタン301

³⁰¹遂 ³¹⁴進 ³¹⁴逆 ³¹⁶連 ³²²遵 ³²⁸送 ³³³速 ³³⁹返 ³³⁹逸 ³⁴⁰達

図9-2 パリフェラル特徴認識結果 (尺度L1)

私は遂を逆と読みます 左ボタン66

⁶⁰逆 ⁶²遁 ⁶⁶遂 ⁶⁶迹 ⁷¹走 ⁷¹連 ⁷²迷 ⁷²速 ⁷²甫 ⁷²兆

図9-3 メッシュ特徴認識結果 (尺度L1)

3. 4 フォントの変形

作成したフォントに対して、自動中間パターン作成機能により任意の2個のパターン間の中間パターンを作成することができます。この機能は「後処理」モードで「文字変形」を選択することによって利用でき、文字変形の詳細アルゴリズムは付録2に示します。またこの機能は文字パターンだけでなく全ての2値画像パターンに適用できます。この機能を使用して作成した中間パターンの例を図10・11に示します。

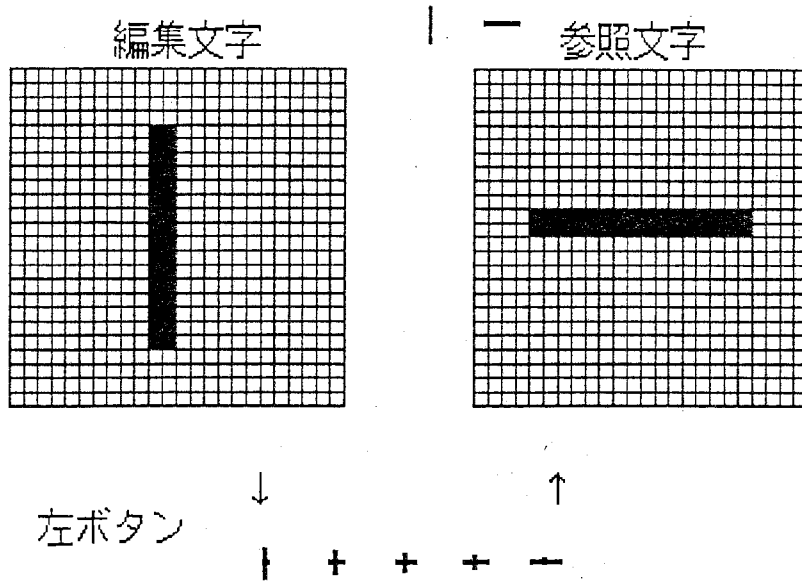


図10 パターン変形例

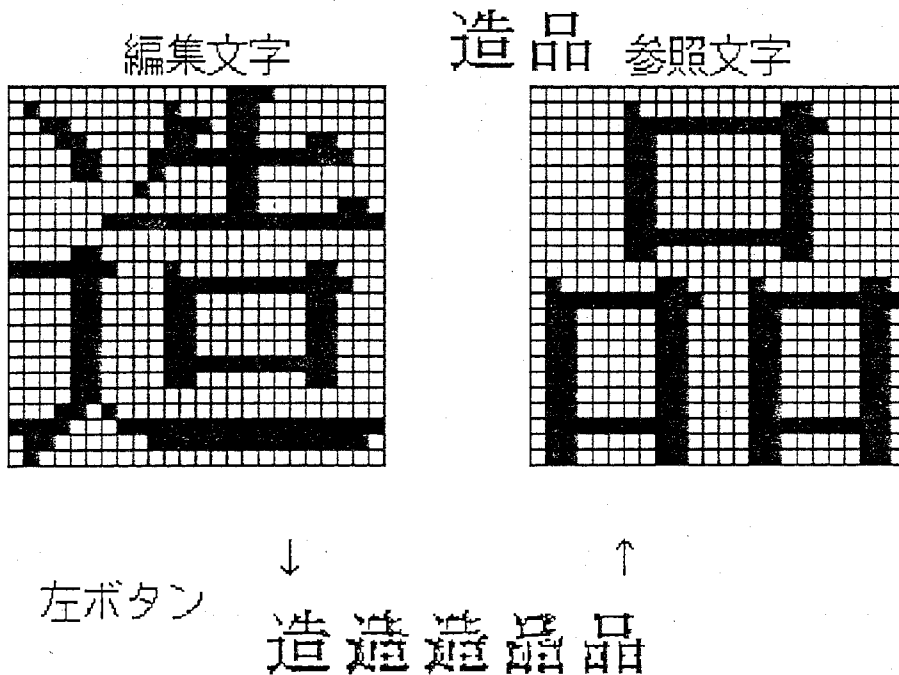


図11 文字変形例

3. 5 作成フォントの入出力

本エディターにより作成したフォントは、MS-DOSのファイルとして保存しています。また、プログラム動作中は100文字分のデータをVRAMに保持しています。作成フォントファイルは日本語ワードプロセッサ（一太郎）の外字として利用可能です。作成フォントをフォントファイルに保存するためには、「後処理」モードで「文字登録」を選択し、登録する位置をマウスによって決めることにより可能です。また、既登録文字フォントを修正するには「外部入力」モードで「文字一覧」を選択し、修正する文字フォント位置を指定することによって可能です。図12はフォント入出力時に表示される画面の例です。



図12 フォント入出力の例

3. 6 その他の機能

前節までの機能の他に次のような機能があります

編集画面の印刷（「後処理」モードで「画面印刷」を選択）

編集文字と参照文字の交換（メニューモードで「1←→2」を選択）

編集文字を参照文字へ複写（メニューモードで「1->2」を選択）

また、文字保存ファイルの一覧印刷は文字入出力画面で出力される、「ハードコピーしますか」に「Y」と入力することにより行えます。

4. フェーブルの使用例

フェーブルを文字認識研究に利用した2つの例を以下に示します。

4. 1 文字認識法の比較

フェーブルは、これまでに文字認識法として提案されている3つの認識特徴、3つの認識尺度をそれぞれ自由に選択して、認識法の特徴を調べることができます。そこで、図13に示すようなJISフォントの漢字“横”の認識実験を基に、文字認識法の比較を行いました。その結果、以下のような点が明らかになりました。

① 9種類の認識結果で10位までの候補がまったく同じ漢字集合であるようなものはない。

② 同じ認識特徴を用いても、認識尺度が異なれば認識順位ばかりでなく、候補となる漢字も違う。従って、認識法のよしあしには抽出する認識特徴と認識尺度の双方の検証が必要である。

③ “横”以外に共通して候補に挙がっている漢字は“構”しかない。また、線密度特徴・類似度の場合の“趨”のように人間にとって思いがけない候補が挙ってくる。

④ ペリフェラル特徴によれば、どの認識尺度でも、候補の漢字が木へんの文字となっている。これはペリフェラル特徴が外形を反映した特徴である為と考えられる。

⑤ ペリフェラル特徴とメッシュ特徴の結果を比較すると、どんな認識尺度を選択しても、“横”と“構”以外共通候補がない。従って、2つの特徴の独立性は高く、両特徴の併用効果が大きいことが予想される。

以上は、1文字の認識結果からの考察でありかなり乱暴な結論ですが、このような形で様々な文字パターンについて認識実験が行われれば、必要とされる文字認識技術が少しずつ明らかになるのではないのでしょうか。

4. 2 変形文字の認識

フェーブルは、2つの漢字の中間パターンを簡単に作成することができます。したがって、文字認識法で大きな問題となる類似漢字の識別について、この中間パターンを用いた認識実験によって検討することがで

きます。例えば、漢字”金”と”全”の中間パターンとして得られる図14の2パターンがどちらに認識されるのが妥当かといった問題を考えてみます。

妥当性の基準として人間の判断基準を利用してみることにします。心理実験によって、図14に示す2パターンが漢字”金”に類似しているか、”全”に類似しているかを7段階評定（”金” = 7、”全” = 1）させました。被験者は8名です。

心理実験の結果を平均すると、パターン1（図14の編集文字）に対して4.63、パターン2（図14の参照文字）に対して2.38という結果を得ました。すなわち、人間はパターン1は”金”に類似し、パターン2は”全”に類似していると評価したことになります。これと同じ結果が得られるような認識法が人間の判断基準に近い認識法ということになります。

4.1と同じように、3つの認識特徴、3つの認識尺度をそれぞれ選択した9種類の認識法で認識実験を行いました。図14は認識特徴がペリフェラル特徴、認識尺度がL1距離の場合のパターン1に対する認識結果です。9種類の認識実験の結果、パターン1が”金”、パターン2が”全”と認識されるのは、ペリフェラル特徴でL1距離、L2距離のときと、メッシュ特徴ですべての認識尺度のときでした。但し、メッシュ特徴の場合パターン2に対する認識結果の第2位はいずれの認識尺度でも”金”ではなく、”介”でした。

このように、高精度の認識法を開発するには、今後共より人間の認識基準に近い認識法の追及が必要と考えます。

横⁰ 構⁷² 検⁷⁵ 搔⁷⁶ 櫛⁷⁶ 梗⁸¹ 模⁸² 槓⁸⁶ 榷⁸⁹ 搜⁸⁹

横⁰ 構¹⁵ 検¹⁶ 搔¹⁶ 梗¹⁷ 模¹⁷ 榷¹⁸ 搜¹⁸ 樟¹⁹ 樺¹⁹

1000横⁰ 990枝 987構 986梗 986技 986検 985趨 985搔 983梓 983桂

線密度特徴文字認識

横⁰ 176構 188検 196榎 200模 230樽 231棋 237梗 248椿 249橡

横⁰ 44検 44榎 44模 46構 48樽 52橡 52榎 53椿 53棟

1000横⁰ 983検 983榎 982構 982模 981校 979樟 979樽 979棟 979梗

ペリフェラル特徴文字認識

横⁰ 31構 37槽 44糟 46描 46槓 47棉 48楢 51楠 51糧

横⁰ 9構 9槽 10糟 10描 11犧 11槓 12惜 12積 12措

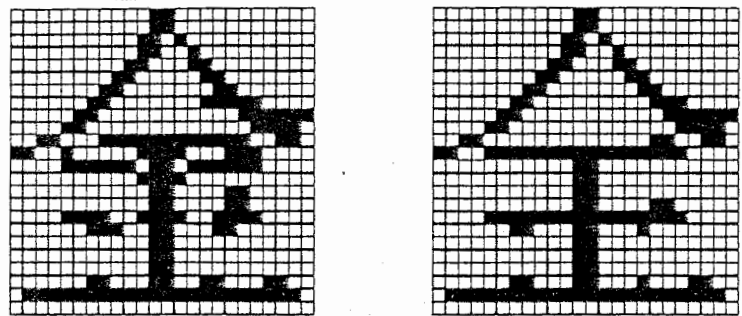
1001横⁰ 980構 976槽 974糟 969描 967槓 965犧 964棉 961槽 961暢

メッシュ特徴文字認識

図 1 3 漢字“横”に対する文字認識結果

(各々の認識尺度 上段：L 1, 中段：L 2, 下段：類似度)

フォントエディター Fable
 左ボタン: ON OFF
 類似評価 線密度 パリフェラル メッシュ ESC
 L1距離 L2距離 類似度
 編集文字 金 全 参照文字



人生は長いと思いません

私は 金 を 金 と読みます 左ボタン202

192 202 334 380 403 421 425 435 447
金 全 企 会 里 垂 食 史 瓦

図 1 4 文字認識例 (金と全から作成した中間文字について)

5. むすび

フェーブルはマウスを使用することにより、よりよいフォント作成環境を目指して作成したものです。また、フォントの入力にマウスからだけでなくかな漢字変換（A T O K 5）及び画像データ入力が可能であり非常に簡単に希望とするフォントが作成できます。動作速度についても、PC 9801のハードウェアをOS（MS-DOS）を介さないでアクセスしているのだからかなり高速です。しかし、その反面 PC 9801以外の機種への移植性に問題があると思います。本エディターの利用分野として、文字のデザイン評価、文字認識機械の開発、他に、マンマシンインタフェイス研究のための視覚刺激作成ツールとしても有望と考えられます。特に、フェーブルが文字認識手法の開発に活用されよりよい手法が確立されることを希望する。

謝辞 ハードウェアについて調査して戴いた A T R 視聴覚研究所
駒木根 研究員
及び、視聴覚機構研究所の諸氏に感謝します。

参考文献

- 1) Schildt, "advanced Turbo Pascal", Osborne McGraw-Hill
- 2) 鷹野 獵場, "骨までしゃぶろう A T O K 5", No.46, The Basic
- 3) " μ PD7220の活用", Jul.1986, プロセッサ, 技術評論社
- 4) " JISC6234(ドットプリンタ用24ドット字形)FD説明書,(財)日本規格協会
- 5) 橋本 編著, "文字認識概論", 電気通信協会

プログラムの起動方法と条件

fable が動作するために必要なファイル

- | | | |
|----|-------------|-----------------------|
| ① | fable.com | 本体 |
| ② | jis6234.dat | J I S フォントファイル |
| ②' | atok.dic | 一太郎の辞書ファイル |
| ③ | henkei.dat | 作成フォント登録ファイル |
| ④ | peiphe.dat | ペリフェラル特徴辞書 |
| ⑤ | senmitu.dat | 線密度特徴辞書 |
| ⑥ | messyu.dat | メッシュ特徴辞書 |
| ⑦ | XXXX.dat | 「画像入力」機能を動作させる場合に必要画像 |

- ①②③ は 絶対に必要なファイル
 ②' は かな漢字変換に必要なファイル
 ④⑤⑥ は 文字認識を動作させるために必要な辞書
 ⑦ は 画像入力ユーティリティ [yomuzo] で作成した画像ファイル

上記プログラムは P C - 9 8 0 1 の M S - D O S (V e r . 3 . 1)
 上で動作します。

[環境]

- ・④⑤⑥ は R A M ディスクに置いた方が認識速度が向上します。
- ・かな漢字変換を使用するためには atok5a.sys atok5b.sys atok.dic
 が必要です。これらのファイルは一太郎を買うともれなく付いてきます。
- ・マウスを使用しているので、マウスドライバを組み込んで下さい。具
 体的には、config.sysに DEVICE=mouse.sys を付け加えるだけです。
- ・グラフィックを使用するために、graph.com が必要です。graph.com
 は本プログラムにもれなく付いてきますので、P C - 9 8 0 1 の立ち上
 げ時に 1 度実行して下さい。graph.comの作者に感謝しましょう！

変形パターンの作成法

2つの2値図形を $f_0(x, y)$ と $f_{n+1}(x, y)$ とし、総ての画素は0もしくは1の値をとるものとする。この2つの図形の n 個の中間パターン $f_i(x, y)$, $i=1, n$ を以下のように定義する。

$$f_i(x, y) = \begin{cases} 1 & \left(\frac{(n+i)x_0 - i x_{n+1}}{n}, \frac{(n+i)y_0 - i y_{n+1}}{n} \right) \\ \text{else} & \\ 0 & \end{cases}$$

但し、 $x_0, y_0, x_{n+1}, y_{n+1}$ は以下の2つの条件のうち何れかを満たす。

1) $f_0(x, y) = f_{n+1}(x_{n+1}, y_{n+1}) = 1$ を満たす任意の (x, y) のうち、 (x_0, y_0) は (x_{n+1}, y_{n+1}) からの距離 $(x - x_{n+1})^2 + (y - y_{n+1})^2$ が最小である。

2) $f_0(x_0, y_0) = f_{n+1}(x, y) = 1$ を満たす任意の (x, y) のうち、 (x_{n+1}, y_{n+1}) は (x_0, y_0) からの距離 $(x_0 - x)^2 + (y_0 - y)^2$ が最小である。

```

( this file is JIS6234(24*24) edit      program )
(      avail graphic routine      made in KIT1 )
(      copy from fontedt.pas      1986.10.14 )
(      1986.11.30 )
(      1986.12.03 )
(      character recog +      1986.12.08 )
(      PC-IN503 data get +      1987.03.06 )
(      ATOK AUTO START +      1987.03.13 )
(      messyu recog +      1987.04.27 )
($U-)
($I pset_hi.lib)      ( V-RAM direct and xor LINE plus )
($I esc_seq.lib)      ( for escape code to CRT )
($I graph.lib)        ( for graphic )
($I kanji.lib)        ( for KANJI to VRAM )
($I mouse.lib)        ( for mouse )
($I hardc.lib)        ( for hard copy )
($I for_atok.lib)     ( for ATOK5 call without CTRL-XFER key )
const
  DEBUG = 0 ;          (* DEBUG FLAG *)
  CELL = 6 ;           (* DOT PATTERN MIN WIDTH      6 *)
  LEL = 100;           (* LEFT REF.PATTERN X LOCATION 100 *)
  LER = 300;           (* RIGHT REF.PATTERN X LOCATION 300 *)
  UP = 100;            (* HEAD LOCATIONS OF 2 PATTERNS *)
  OFFSET = 2 ;        (* FOR PAINT *)
  ENDX = 142;         (* MENU LOCATION *)
  SAVEX = 222;        (* MENU LOCATION *)
  LOOKX = 302;        (* MENU LOCATION *)
  OPTX = 382;         (* MENU LOCATION *)
  OPT2X = 462;        (* MENU LOCATION *)
  ON_OFF = 0;
  ON_ON = 1;
  OFF_OFF = 2;
  L1 = 1;
  L2 = 2;
  RUI = 3;

type
  fontbuf = array[1..128] of byte ;
  menuyou = string[120] ;
  gaibuf = array[1..2,0..23,0..23] of integer;
  sortbuf = array[0..3500] of integer;
var
  buf : getbuf ; (* 128 bytes read buffer *)
  tmp : byte ; (* temporary *)
  tmp2 : integer ; (* temporary *)
  tmp1 : integer ; (* temporary *)
  x,y : integer ; (* x - y absolute location *)
  px,py : integer ; (* select menu x location *)
  i,j,k,l,m : integer ; (* loop counter *)
  fp,fq,fr : file ; (* file pointer *)
  ans : string[4] ; (* answer buffer*)
  gdata : array[1..5,0..400] of byte; (* get put bufer *)
  cell_dat : array[0..400] of byte; (* cell paint data *)
  dell_dat : array[0..400] of byte; (* cell paint data *)
  xx : array[1..5] of integer; (* put x-location of get data *)
  yy : array[1..5] of integer; (* put y-location of get data *)
  gaiji_dat : gaibuf ; (* gaiji data *)
  wdata : fontbuf ; (* font data buffer *)
  rmb,lmb : integer ; (* mouse botton *)
  iro : integer ; (* font hyouji color *)
  pass_f : byte ; (* redraw flag 0:yes 1:no *)
  last_page : byte ;
  koko,koko2 : integer ;
  hide : integer ;
  hide2 : integer ;
  syakudo : integer ;
  ue,lue,yoko,rue: byte ;
  moji_dat : array[0..299,0..127] of byte;
  mo_dat : array[0..299,0..37] of byte;
  ans1 : array[0..20] of char ; (* get key buffer *)

($I fabel1.pas)
procedure f_save;
var
  moji_sum : integer ; (* touroku moji sum *)
  page_sum : integer ; (* touroku moji sum/200 *)
  now_page : integer ; (* now look page *)
  max_x,max_y : integer ; (* save max location *)
  tx,ty : integer ;
  mx,my : integer ;
begin
  mou_flush;
  mousexrange(200,344);
  mouseyrange(70,324);
  mousepset(270,70);
  repeat
    mousepget(x,y,lmb,rmb);
    if (y<300)
      then
        begin
          mx := LER;
          my := UP-30;
        end;
    if (x<300)and (y<300)
      then
        begin
          mx := LER-30;
          my := UP-30;
        end;
    if (200<=x)and (x <230) and (y)=300)
      then

```

```

begin
  mx := 200;
  my := 300;
end;
if (230<=x)and (x <260) and (y)=300)
then
begin
  mx := 230;
  my := 300;
end;
if (260<=x)and (x <290) and (y)=300)
then
begin
  mx := 260;
  my := 300;
end;
if (290<=x)and (x <320) and (y)=300)
then
begin
  mx := 290;
  my := 300;
end;
if (320<=x)and (x <350) and (y)=300)
then
begin
  mx := 320;
  my := 300;
end;
line(mx-1,my-1,mx+24,my+24,4,1,0,0,0,0,0);
delay(200);
line(mx-1,my-1,mx+24,my+24,0,1,0,0,0,0,0);
until (lmb = 1);
mou_flush;
makewdata(mx,my,wdata);
get(mx,my,mx+23,my+23,ofs(gdata[3][0]),seg(gdata[3][0]),230);
mousexrange(0,639);
mouseyrange(0,399);
port[$a6] := 1;
port[$a4] := 1;
reset(fq);
moji_sum := filesize(fq);
page_sum := (moji_sum div 100) + 1;
now_page := last_page;
max_x := ((moji_sum mod 100) mod 20);
max_y := ((moji_sum mod 100) div 20);
mousepset(0,0);
mousexrange(0,99);
mouseyrange(0,24);
rmb := 1;
clrscr;
if (now_page <> 0) then
begin
  gotoxy(10,1);
  writeln('登録文字 一覧表 左ボタン:登録 右ボタン:ページ');
  tmp1 := (now_page -1)*100;
  for i:= tmp1+1 to tmp1+100 do
  begin
    gotoxy(((i-1) mod 20)*4+1,2+((i-1-tmp1) div 20)*3);
    write(i:3);
  end;
end;
repeat
if last_page = 0 then
begin
if rmb = 1 then
begin
now_page := now_page + 1;
if now_page > page_sum
then now_page := 1;
tmp1 := (now_page -1)*100;
cls;
clrscr;
gotoxy(10,1);
writeln('登録文字 一覧表 左ボタン:登録 右ボタン:ページ');
for i:= tmp1+1 to tmp1+100 do
begin
gotoxy(((i-1) mod 20)*4+1,2+((i-1-tmp1) div 20)*3);
write(i:3);
end;
seek(fq,(now_page-1)*100);
tmp2 := 100;
if (now_page = page_sum)
then tmp2 := (moji_sum mod 100);
for i:= 1 to tmp2 do
begin
blockread(fq,buf,1,tmp1);
kanp24(((i-1) mod 20)*32+1,32+48*(((i-1) div 20),7,buf);
end;
mou_flush;
last_page := now_page;
end;
end;
tx := (x div 5);
ty := (y div 5);
if (now_page = page_sum)
then
if ((ty >max_y) or ((ty = max_y) and (tx>= max_x)))
then
begin
tx := max_x;
ty := max_y;

```



```

        if tx = 20 then
            begin
                tx := 0;
                ty := max_y + 1;
            end;
            mousepset(tx*5+1,ty*5+1);
        end;
        line(tx*32,ty*48+31,tx*32+25,ty*48+57,3,1,0,0,0,0,0,0);
        delay(100);
        line(tx*32,ty*48+31,tx*32+25,ty*48+57,0,1,0,0,0,0,0,0);
        mousepget(x,y,lmb,rmb);
        if (lmb = 1) then
            begin
                reset(fq);
                seek(fq,ty*20+tx+(now_page-1)*100);
                blockwrite(fq,wdata,1,tmp1);
                {
                    gaiji_get(tx*32+1,ty*48+32,1,gaiji_dat);
                    get(tx*32+1,ty*48+32,tx*32+24,ty*48+55,ofs(gdata[1][0]),seg(gdata[1][0]),230);
                }
                put(tx*32+1,ty*48+32,ofs(gdata[3][0]),seg(gdata[3][0]),230,0,0,0,0);
            end;
            if (rmb = 1) then
                last_page := 0;
        until (lmb = 1);
        mousexrange(0,639);
        mouseyrange(0,399);
        reset(fq);
        seek(fq,filesize(fq));
        port[$a6] := 0; (* 描画アクセス プレーン1 *)
        cls;
        clrscr;
        port[$a4] := 0; (* 表示アクセス プレーン1 *)
    end;

procedure f_look;
var
    moji_sum      : integer          ; (* touroku moji sum *)
    page_sum      : integer          ; (* touroku moji sum/200 *)
    now_page      : integer          ; (* now look page *)
    max_x,max_y   : integer          ; (* save max location *)
    tx,ty         : integer          ;
begin
    port[$a4] := 1;
    port[$a6] := 1;
    reset(fq);
    moji_sum := filesize(fq);
    page_sum := (moji_sum div 100) + 1;
    max_x := ((moji_sum mod 100) mod 20);
    max_y := ((moji_sum mod 100) div 20);
    now_page := last_page;
    mousepset(0,0);
    mousexrange(0,99);
    mouseyrange(0,24);
    rmb := 1;
    clrscr;
    if now_page <> 0 then
        begin
            gotoxy(10,1);
            writeln('登録文字 一覧表 左ボタン:EDIT 右ボタン:ページ');
            tmp1 := (now_page - 1)*100;
            for i:= tmp1+1 to tmp1+100 do
                begin
                    gotoxy(((i-1) mod 20)*4+1,2+((i-1-tmp1) div 20)*3);
                    write(i:3);
                end;
            end;
        repeat
            if last_page = 0 then
                begin
                    if rmb = 1 then
                        begin
                            now_page := now_page + 1;
                            if now_page > page_sum
                                then now_page := 1;
                            tmp1 := (now_page - 1)*100;
                            cls;
                            clrscr;
                            gotoxy(10,1);
                            writeln('登録文字 一覧表 左ボタン:EDIT 右ボタン:ページ');
                            for i:= tmp1+1 to tmp1+100 do
                                begin
                                    write('',2+((i-1-tmp1) div 20)*3,',',(i-1) mod 20)*4+1,'H',i:3);
                                    {
                                        gotoxy(((i-1) mod 20)*4+1,2+((i-1-tmp1) div 20)*3);
                                        write(i:3);
                                    }
                                end;
                            seek(fq,(now_page-1)*100);
                            tmp2 := 100;
                            if (now_page = page_sum)
                                then tmp2 := (moji_sum mod 100);
                            for i:= 1 to tmp2 do
                                begin
                                    blockread(fq,buf,1,tmp1);
                                    kanp24(((i-1) mod 20)*32+1,32+48*((i-1) div 20),7,buf);
                                end;
                            mou_flush;
                            last_page := now_page;
                        end;
                    end;
                end;
            end;
end;
end;

```

```

tx := (x div 5);
ty := (y div 5);
if (now_page = page_sum)
then
  if ((ty > max_y) or ((ty = max_y) and (tx >= max_x)))
  then
    begin
      tx := max_x;
      ty := max_y;
      if tx = 20 then
        begin
          tx := 0;
          ty := max_y + 1;
        end;
      mousepset(tx*5+1,ty*5+1);
    end;
  line(tx*32,ty*48+31,tx*32+25,ty*48+57,3,1,0,0,0,0,0,0);
  delay(100);
  line(tx*32,ty*48+31,tx*32+25,ty*48+57,0,1,0,0,0,0,0,0);
  mousepget(x,y,lmb,rmb);
  if (lmb = 1) then
    begin
      gaiji_get(tx*32+1,ty*48+32,1,gaiji_dat);
      get(tx*32+1,ty*48+32,tx*32+24,ty*48+55,ofs(gdata[1][0]),seg(gdata[1][0]),230);
    end;
  if (rmb = 1) then
    last_page := 0;
until (lmb = 1);
mouseyrange(0,399);
reset(fq);
seek(fq,filesize(fq));
mou_flush;
gotoxy(1,1);
textcolor(2);
writeln('<<<< ハードコピー>>>しますか? 左ボタン:No 右ボタン:Yes');
repeat
  mousepget(x,y,lmb,rmb);
  if(rmb = 1)
  then
    begin
      gotoxy(10,20);
      writeln('約2分お待ち下さい');
      hard_copy_screen;
    end;
until (lmb = 1) or (rmb = 1);
textcolor(7);
mou_flush;
port[$a6] := 0;
cls;
clrscr;
port[$a4] := 0;
end;
procedure font_edit;
var
  calx,caly : integer ; (* temporary *)
  cell_x,cell_y : integer ; (* cell location*)
  px : integer ; (* select menu *)
  on_mode : integer ; (* successive on *)
  hide3 : integer ;
begin
  pass_f := 0;
  on_mode := 0;
  mousexrange(LEL,LEL+CELL*24-1);
  mouseyrange(UP-1,UP+CELL*24-1);
  mousepset(LEL+CELL*2,UP+CELL*2);
  mousesens(4,4);
  repeat (* フォントエディター *)
    mousepget(x,y,lmb,rmb);
    if (y = (UP-1)) and (x > LEL) then
      y := UP;
    if (y = (UP - 1)) and (x = LEL)
    then
      begin
        px := menu_ha(' [モード] 連続ON ノーマル 連OFF 危険です 鼠感覚');
        case px of
          ENDX:
            begin
              on_mode := ON_ON;
            end;
          SAVEX:
            begin
              on_mode := ON_OFF;
            end;
          LOOKX:
            begin
              on_mode := OFF_OFF;
            end;
          OPTX:
            begin
              for i := 7 downto 0 do
                begin
                  gotoxy(1,18);
                  esc_seq(REV_BL,37-i);
                  writeln(' ');
                  writeln(' 作者は 刺客研究室の ');
                  writeln(' Morikiti ');
                  writeln(' と 死角研究室の ');
                  writeln(' Yokosawaです ');
                  writeln(' ');
                  gotoxy(1,18);
                end;
              end;
            end;
        end;
      end;
    end;
  end;
end;

```

```

delay(800);
writeln('');
writeln('');
writeln('');
writeln('');
writeln('');
writeln('');
writeln('');
delay(500);
end;
esc_seq(RESET,0);
textcolor(7);
end;
OPT2X:
begin
  if hide = 1
    then
      hide := 2
    else
      hide := 1;
      hide3 := 1;
      esc_seq(REV_BL, GREEN);
      write('MOUSE INTERVAL [0-3] ');
      readln(hide3);
      esc_seq(RESET,0);
      esc_seq(REV_IND,1);
      write(' ');
      port[$bfdb] := hide3;
    end;

  end;

  gotoxy(1,3);
  writeln('');
  writeln('');
  mousexrange(LEL,LEL+CELL*24-1);
  mouseyrange(UP-1,UP+CELL*24-1);
end
else
begin
  calx := ((x - LEL) div CELL);
  cell_x := calx*CELL + LEL;
  caly := ((y - UP) div CELL);
  cell_y := caly*CELL + UP;
  line(cell_x,cell_y,cell_x+CELL,cell_y+CELL,4,1,0,0,0,0,0,0);
  line(calx*CELL+LER,cell_y,calx*CELL+LER+CELL,cell_y+CELL,4,1,0,0,0,0,0,0);
  ( delay(100); )
  line(cell_x,cell_y,cell_x+CELL,cell_y+CELL,1,1,0,0,0,0,0,0);
  line(calx*CELL+LER,cell_y,calx*CELL+LER+CELL,cell_y+CELL,1,1,0,0,0,0,0,0);
  if lmb = 1
    then
      begin
        tmp1 := point(cell_x+1,cell_y+1);
        case on_mode of
          ON_OFF:
            begin
              if tmp1 > 0
                then
                  begin
                    paint(cell_x+1,cell_y+1,0,1);
                    preset(LER-30+calx,UP-30+caly);
                  end
                else
                  begin
                    paint(cell_x+1,cell_y+1,iro,1);
                    pset(LER-30+calx,UP-30+caly,iro);
                  end;
                mou_flush;
              end;
            ON_ON:
              begin
                if tmp1 = 0
                  then
                    begin
                      paint(cell_x+1,cell_y+1,iro,1);
                      pset(LER-30+calx,UP-30+caly,iro);
                    end;
                end;
            OFF_OFF:
              begin
                if tmp1 > 0
                  then
                    begin
                      paint(cell_x+1,cell_y+1,0,1);
                      preset(LER-30+calx,UP-30+caly);
                    end;
                end;
            end; (* case end *)
          end; (* if end *)
        end; (* else end *)
      until (rmb = 1);

      mou_flush;
      makewdata(LER-30,UP-30,wdata);
      mousesens(8,8);
    end;
  procedure font_youi;
  begin
    cis;
    for x:=0 to 24 do
      begin

```

<お知らせ>として
 ユーティリティを
 フォントを
 イメージを読み
 ご利用下さい (別売り)

右ボタン: MENU

```

line(LEL+CELL*x,UP,LEL+CELL*x,UP+CELL*24,1,0,0,0,0,0,0);
line(LER+CELL*x,UP,LER+CELL*x,UP+CELL*24,1,0,0,0,0,0,0);
end;
for y:=0 to 24 do
begin
line(LEL,UP+CELL*y,LEL+CELL*24,UP+CELL*y,1,0,0,0,0,0,0);
line(LER,UP+CELL*y,LER+CELL*24,UP+CELL*y,1,0,0,0,0,0,0);
end;
get(LEL,UP,LEL+CELL-1,UP+CELL-1,ofs(dell_dat[0]),seg(dell_dat[0]),50);
paint(LEL+OFSET,UP+OFSET,7,1);
get(LEL,UP,LEL+CELL-1,UP+CELL-1,ofs(cell_dat[0]),seg(cell_dat[0]),50);
paint(LEL+OFSET,UP+OFSET,0,1);
for x:=0 to 23 do
for y:=0 to 23 do
begin
tmp2 := gaiji_dat[1][x][y];
if tmp2 > 0
then
begin
put(LEL+CELL*x,UP+CELL*y,ofs(cell_dat[0]),seg(cell_dat[0]),50,0,0,0,0);
end;
end;
put(LER-30,UP-30,ofs(gdata[1][0]),seg(gdata[1][0]),230,0,0,0,0);
for x:=0 to 23 do
for y:=0 to 23 do
begin
tmp2 := gaiji_dat[2][x][y];
if tmp2 > 0
then
put(LER+CELL*x,UP+CELL*y,ofs(cell_dat[0]),seg(cell_dat[0]),50,0,0,0,0);
end;
put(LER,UP-30,ofs(gdata[2][0]),seg(gdata[2][0]),230,0,0,0,0);
end;
end;

```

```

procedure com_hyo;

```

```

var

```

```

hik_dat:array[0..23,0..23] of integer ;
x,y,tmp2 : integer ;

```

```

begin

```

```

gaiji_get(LER-30,UP-30,1,gaiji_dat);
for x := 0 to 23 do
for y:= 0 to 23 do
begin
hik_dat[x][y] := 0;
if((gaiji_dat[1][x][y]=7) and (gaiji_dat[2][x][y]=7))
then hik_dat[x][y] := 7;
if((gaiji_dat[1][x][y]=7) and (gaiji_dat[2][x][y]=0))
then hik_dat[x][y] := 2;
if((gaiji_dat[1][x][y]=0) and (gaiji_dat[2][x][y]=7))
then hik_dat[x][y] := 4;
end;

```

```

for x:=0 to 23 do
for y:=0 to 23 do
begin
tmp2 := hik_dat[x][y];
if tmp2 = 2
then
paint(LEL+CELL*x+OFSET,UP+CELL*y+OFSET,tmp2,1);
if tmp2 = 4
then
paint(LER+CELL*x+OFSET,UP+CELL*y+OFSET,tmp2,1);
end;
end;

```

```

end;

```

```

($I recog.pas)

```

```

begin (* ここからメインルーチン *)

```

```

syokika :=
hide := 2;
hide2 := 0;
koko2 := 0;
ue := 1;
lue := 1;
rue := 1;
yoko := 1;
repeat

```

```

ans[1] := 'n';
clrscr;
mou_flush;
writeln('
writeln('
writeln('
writeln('
writeln('');
writeln('');
if(pass_f = 0) then
begin
font_youi;
end;
font_edit;
px := menu_ha(' [メニュー] 1 → 2 1 ← 2 外部入力 後処理 終了 ');
gaiji_get(LER-30,UP-30,1,gaiji_dat);
get(LER-30,UP-30,LER-30+23,UP-30+23,ofs(gdata[1][0]),seg(gdata[1][0]),230);
case px of
ENDX:
begin
mv_12;
end;
SAVEX:
begin
cp_12;

```

```

フロントエディター Fable ');
左ボタン: ON OFF');
右ボタン: MENU ');

```

```

編集文字 参照文字');
(* 文字パターンを書き直さなくてもよい場合 *)

```

```

end;
LOOKX:
begin
px := menu_ha(' [外部入力] 文字一覧 文字入力 画像入力 ESC ');
case px of
ENDX:
begin
f_look;
end;
SAVEX:
begin
kan_input(hide);
(
hard_copy_screen;
)
end;
LOOKX:
begin
gazoget;
end;
OPTX:
begin
pass_f := 1;
end;
OPT2X:
begin
pass_f := 1;
end;
end; (* option menu 1 case end *)
end;
OPTX:
begin
px := menu_ha(' [後処理] 文字登録 画面印刷 文字認識 文字変形 ESC ');
case px of
ENDX:
begin
f_save;
end;
SAVEX:
begin
hard_copy_screen;
end;
LOOKX:
begin
repeat
pass_f := 3;
px := menu_ha2(' [文字認識] 類似評価 線密度 解像度 メッシュ ESC ');
: [認識尺度] L1距離 L2距離 類似度 ');
(* CAUTION external value syakudo is changed in this sub-routine *)
case px of
ENDX:
begin
com_hyo;
tan_ruiji;
periphe;
end;
SAVEX:
begin
moji_nin(3);
end;
LOOKX:
begin
moji_nin(2);
end;
OPTX:
begin
moji_nin(4);
end;
OPT2X:
begin
pass_f := 1;
end;
end; (* sub menu end *)
font_youi;
for i:= 1 to 10 do
begin
gotoxy(1,17);
delline;
end;
until(pass_f = 1)
end;
OPTX:
begin
henkei;
end;
OPT2X:
begin
pass_f := 1;
end;
end; (* option menu 1 case end *)
end;
OPT2X:
begin
cls;
clrscr;
gotoxy(20,10);
write('左ボタン:戻る ');
esc_seq(REVER, GREEN);
write('右');
esc_seq(RFSET, 0);
write('ボタン:');

```

```
esc_seq(BLINK, GREEN);
write('終了');
esc_seq(RESET, 0);
repeat
    mousepget(x, y, lmb, rmb);
until ((rmb = 1) or (lmb = 1));
if rmb = 1
    then
        ans[1] := 'y'; (* 唯一のメイン脱出 *)
    end;
end; (* main case end *)
until (ans[1] = 'y');
```

(* 終了処理 *)

```
mou_flush;
mouseerase;
cursor_on;
close(fp);
close(fq);
clrscr;
cls;
```

nd.

```

procedure mou_flush;          (* 余分に押されたボタンのクリア *)
begin
  while (lmb > 0) or (rmb > 0)
  begin
    mousepget(x,y,lmb,rmb);
  end;
end;
procedure gaiji_get(sx,sy_loc:integer;var gaiji_dat:gaibuf);
  (* 文字データ作成 *)
var
  k,j : integer ; (* counter *)
begin
  for k:=sx to sx+23 do
  begin
    for j:= sy to sy+23 do
    begin
      gaiji_dat[loc][k-sx][j-sy] := point(k,j);
    end;
  end;
end;
function menu_ha(menu_moji:menuyou):integer;
var
  mx : integer ; (* menu x location *)
  x,y,lmb,rmb: integer ;
label OYA;
begin
  gotoxy(3,3);
  mousexrange(115,155);
  mouseyrange(100,130);
  mousepset(130,100);

  writeln(menu_moji);
  writeln('左ボタンで選択');
  repeat
    mousepget(x,y,lmb,rmb);
    if y > 125 then goto OYA;
    if x < 125
      then mx := ENDX;
    if (125<=x)and (x < 135)
      then mx := SAVEX;
    if (135<=x)and (x < 145)
      then mx := LOOKX;
    if (145<=x)and (x <= 155)
      then mx := OPTX;
    if (155<=x) and (x <= 165)
      then mx := OPT2X;
    line(mx,25,mx+65,50,4,1,0,0,0,0,0);
    delay(200);
    line(mx,25,mx+65,50,0,1,0,0,0,0,0);
  until (lmb = 1);
OYA:
  mousexrange(0,639);
  mouseyrange(0,399);
  mou_flush;
  menu_ha := mx;
end;
function menu_ha2(menu_moji,menu_moji2:menuyou):integer;
var
  mx          : integer ; (* menu x location *)
  mmx         : integer ;
  x,y,lmb,rmb : integer ;
  y_off       : integer ; (* 2 danme offset *)
  syaku_ok,syuhou_ok : integer ;
  syu_x,sya_x : integer ;
  syu_x2,sya_x2 : integer ;
const
  OK = 1;
begin
  mousexrange(115,155);
  mouseyrange(100,130);
  mousepset(130,100);
  syaku_ok := 0;
  syuhou_ok := 0;
  sya_x := 0;
  syu_x := 0;
  sya_x2 := 0;
  syu_x2 := 0;
  gotoxy(3,3);
  writeln(menu_moji);
  gotoxy(1,4);
  writeln(' ');
  gotoxy(3,4);
  writeln(menu_moji2);
  esc_seq(REVER,WHITE);
  writeln('左ボタンで選択');
  esc_seq(RESET,0);
  repeat
    mousepget(x,y,lmb,rmb);
    if (y < 125) then
      begin
        y_off := 0;
        if x < 125
          then mx := ENDX;
        if (125<=x)and (x < 135)
          then mx := SAVEX;
        if (135<=x)and (x < 145)
          then mx := LOOKX;
        if (145<=x)and (x <= 155)
          then mx := OPTX;
        if (155<=x) and (x <= 165)

```

```

then mx := OPT2X;
if (lmb = 1) then
begin
  syu_x2 := syu_x;
  syu_x := mx;
  mmx := mx;
  if ((mx = ENDX) or (mx = OPT2X)) then
    syaku_ok := OK;
  syuhou_ok := OK;
end;
end;
if (y >= 125) then
begin
  y_off := 19;
  if x < 125
  then
    begin
      mx := SAVEX;
      syakudo := L1;
    end;
  if (125 <= x) and (x < 135)
  then
    begin
      mx := SAVEX;
      syakudo := L1;
    end;
  if (135 <= x) and (x < 145)
  then
    begin
      mx := LOOKX;
      syakudo := L2;
    end;
  if (145 <= x) and (x <= 155)
  then
    begin
      mx := OPTX;
      syakudo := RUI;
    end;
  if (155 <= x) and (x <= 165)
  then mx := OPTX;
  if (lmb = 1) then
    begin
      sya_x2 := sya_x;
      sya_x := mx;
      syaku_ok := OK;
    end;
end;

```

```

line(mx,30+y_off,mx+65,48+y_off,4,1,0,0,0,0,0,0);
delay(200);
line(mx,30+y_off,mx+65,48+y_off,0,1,0,0,0,0,0,0);
if (syu_x > 0) then
  line(syu_x,30,syu_x+65,48,2,1,0,0,0,0,0,0);
if (sya_x > 0) then
  line(sya_x,30+19,sya_x+65,48+19,2,1,0,0,0,0,0,0);
if ((syu_x2 > 0) and (syu_x2 < syu_x)) then
  line(syu_x2,30,syu_x2+65,48,0,1,0,0,0,0,0,0);
if ((sya_x2 > 0) and (sya_x2 < sya_x)) then
  line(sya_x2,30+19,sya_x2+65,48+19,0,1,0,0,0,0,0,0);
until ((syaku_ok = OK) and (syuhou_ok = OK));

```

```

mousexrange(0,639);
mouseyrange(0,399);
mou_flush;
menu_ha2 := mmx;
end;

```

```

procedure makewdata(xx,yy:integer;var wdata:fonthuf);
(* フォントのx-y座標;データ格納バッファ *)

```

```

var
  i,j,k : integer ; (* counter *)
  aru : integer ; (* is there a point ? *)
begin
  for i:=1 to 128 do
    wdata[i] := 0;
  wdata[1] := 1 ;
  wdata[2] := 9 ;
  wdata[3] := 9 ;
  wdata[4] := 9 ;
  wdata[5] := 9 ;

  for i:= 0 to 23 do
    for j:= 1 to 3 do
      for k:= 0 to 7 do
        begin
          aru := 0;
          if point((j-1)*8+k+xx,yy+i) > 0
          then
            aru := 1;
            wdata[i*3+j+54] := (wdata[i*3+j+54] shl 1) + aru;
          end;
        end;
      end;
    end;
  end;
end;

```

```

procedure arimasen(err_num,addr:integer);
(* エラーハンドリング *)

```

```

begin
  if hi(err_num) = 1 then
    begin
      textcolor(6);
      writeln(' ');
      writeln(' J I S文字パターン格納辞書 ');
      writeln(' J I S 6 2 3 4 . d a t ');
      writeln(' 作成文字パターン格納辞書 ');
    end;
  end;
end;

```



```

        writeln('             henkei.dat          ');
        writeln('             上記2つは実行時にディスク指定 ');
        writeln('             文字認識に必要な辞書          ');
        writeln('             senmitu.dat          ');
        writeln('             peiphe.dat          ');
        writeln('             messyu.dat          ');
        writeln('             上記3つはディスクd:に固定 ');
        writeln('             それと、 ');
        writeln('             画像データの読み込みで ');
        writeln('             指定されたファイルが ');
        writeln('             指定されたディスクに存在しません ');
        esc_seq(REVER,YELLOW);
        writeln('             今一度、お手元のディスクを ');
        writeln('             お確かめ下さい！ ');
        writeln('             ');
        esc_seq(RESET,0);
        textcolor(7);
    end;
    if hi(err_num) = 0 then
        writeln(' Ctrl-Cを押しましたね');
    if hi(err_num) = 2 then
        writeln(' バグ情報は内線619まで!!!');
    cursor_on;
end;
procedure tan_ruiji; (* 単純類似度計算 *)
var
ruijido      : real          ;          (* simple similarity *)
lp_sum,rp_sum : integer      ;
and_sum      : integer      ;
begin
    lp_sum := 0;
    rp_sum := 0;
    and_sum := 0;
    for x:=0 to 23 do
        for y:=0 to 23 do
            begin
                if point(LER-30+x,UP-30+y) > 0
                    then
                        lp_sum := lp_sum + 1;
                if point(LER+x,UP-30+y) > 0
                    then
                        rp_sum := rp_sum + 1;
                if (point(LER-30+x,UP-30+y) > 0) and (point(LER+x,UP-30+y) > 0)
                    then
                        and_sum := and_sum + 1;
            end;
            ruijido := sqrt(lp_sum)*sqrt(rp_sum);
            if ruijido > 0
                then
                    ruijido := and_sum / ruijido;
                    gotoxy(20,18);
                    write(' 左パターン ドット数: ');
                    writeln(lp_sum:4);
                    gotoxy(20,19);
                    write(' 右パターン ドット数: ');
                    writeln(rp_sum:4);
                    gotoxy(20,20);
                    write(' 重複 ');
                    writeln(and_sum:4);
                    gotoxy(20,21);
                    write(' 単純類似度 ');
                    writeln(ruijido:1:3);
            end;
        end;
    end;
procedure periphe; (* パリフェラル特徴計算 *)
var
rpu,rpd      : array[0..23] of integer;
rpl,rpr      : array[0..23] of integer;
lpu,lpd      : array[0..23] of integer;
lpl,lpr      : array[0..23] of integer;
rpu2,rpd2    : array[0..23] of integer;
rpl2,rpr2    : array[0..23] of integer;
lpu2,lpd2    : array[0..23] of integer;
lpl2,lpr2    : array[0..23] of integer;
ruijido      : real          ;
lp_sum,rp_sum : real          ;
and_sum      : real          ;
ruijido2     : real          ;
lp_sum2,rp_sum2: real        ;
and_sum2     : real          ;
begin
    for x:= 0 to 23 do (* 上方向から *)
        begin
            y := 0;
            while((point(LER-30+x,UP-30+y) = 0) and (y<24)) do
                y := y + 1;
            lpu[x] := y;
            while((point(LER-30+x,UP-30+y) = 7) and (y<24)) do
                y := y + 1;
            while((point(LER-30+x,UP-30+y) = 0) and (y<24)) do
                y := y + 1;
            lpu2[x] := y;
            y := 0;
            while((point(LER+x,UP-30+y) = 0) and (y<24)) do
                y := y + 1;
            rpu[x] := y;
            while((point(LER+x,UP-30+y) = 7) and (y<24)) do
                y := y + 1;
            while((point(LER+x,UP-30+y) = 0) and (y<24)) do
                y := y + 1;
            rpu2[x] := y;
        end;
    end;
end;

```

```

for x:= 0 to 23 do (* 下方向から *)
begin
y := 23;
while((point(LER-30+x,UP-30+y) = 0) and (y>=0)) do
y := y - 1;
lpd[x] := 23 - y;
while((point(LER-30+x,UP-30+y) = 7) and (y>=0)) do
y := y - 1;
while((point(LER-30+x,UP-30+y) = 0) and (y>=0)) do
y := y - 1;
lpd2[x] := 23 - y;
y := 23;
while((point(LER+x,UP-30+y) = 0) and (y>=0)) do
y := y - 1;
rpd[x] := 23 - y;
while((point(LER+x,UP-30+y) = 7) and (y>=0)) do
y := y - 1;
while((point(LER+x,UP-30+y) = 0) and (y>=0)) do
y := y - 1;
rpd2[x] := 23 - y;
end;
for y:= 0 to 23 do (* 左方向から *)
begin
x := 0;
while((point(LER-30+x,UP-30+y) = 0) and (x<24)) do
x := x + 1;
lpl[y] := x;
while((point(LER-30+x,UP-30+y) = 7) and (x<24)) do
x := x + 1;
while((point(LER-30+x,UP-30+y) = 0) and (x<24)) do
x := x + 1;
lpl2[y] := x;
x := 0;
while((point(LER+x,UP-30+y) = 0) and (x<24)) do
x := x + 1;
rpl[y] := x;
while((point(LER+x,UP-30+y) = 7) and (x<24)) do
x := x + 1;
while((point(LER+x,UP-30+y) = 0) and (x<24)) do
x := x + 1;
rpl2[y] := x;
end;
for y:= 0 to 23 do (* 右方向から *)
begin
x := 23;
while((point(LER-30+x,UP-30+y) = 0) and (x>=0)) do
x := x - 1;

lpr[y] := 23 - x;
while((point(LER-30+x,UP-30+y) = 7) and (x>=0)) do
x := x - 1;
while((point(LER-30+x,UP-30+y) = 0) and (x>=0)) do
x := x - 1;
lpr2[y] := 23 - x;
x := 23;
while((point(LER+x,UP-30+y) = 0) and (x>=0)) do
x := x - 1;
rpr[y] := 23 - x;
while((point(LER+x,UP-30+y) = 7) and (x>=0)) do
x := x - 1;
while((point(LER+x,UP-30+y) = 0) and (x>=0)) do
x := x - 1;
rpr2[y] := 23 - x;
end;
lp_sum := 0;
lp_sum2 := 0;
for x:= 0 to 23 do
begin
lp_sum := lp_sum + lpr[x]*lpr[x]+lpl[x]*lpl[x]+lpu[x]*lpu[x]+lpd[x]*lpd[x];
lp_sum2 := lp_sum2 + lpr2[x]*lpr2[x]+lpl2[x]*lpl2[x]+lpu2[x]*lpu2[x]+lpd2[x]*lpd2[x];
end;
rp_sum := 0;
rp_sum2 := 0;
for x:= 0 to 23 do
begin
rp_sum := rp_sum + rpr[x]*rpr[x]+rpl[x]*rpl[x]+rpu[x]*rpu[x]+rpd[x]*rpd[x];
rp_sum2 := rp_sum2 + rpr2[x]*rpr2[x]+rpl2[x]*rpl2[x]+rpu2[x]*rpu2[x]+rpd2[x]*rpd2[x];
end;
and_sum := 0;
and_sum2 := 0;
for x:=0 to 23 do
begin
and_sum := and_sum + rpr[x]*lpr[x]+rpl[x]*lpl[x]+rpu[x]*lpu[x]+rpd[x]*lpd[x];
and_sum2 := and_sum2 + rpr2[x]*lpr2[x]+rpl2[x]*lpl2[x]+rpu2[x]*lpu2[x]+rpd2[x]*lpd2[x];
end;

ruijido := sqrt(lp_sum) * sqrt(rp_sum);
if ruijido > 0
then
ruijido := and_sum / ruijido;
ruijido2 := sqrt(lp_sum2+lp_sum) * sqrt(rp_sum2+rp_sum);
if ruijido2 > 0
then
ruijido2 := (and_sum+and_sum2) / ruijido2;
gotoxy(20,22);
write('ヘリフェラル類似度 1次:');
write(ruijido:1:3);
write('1+2次:');
writeln(ruijido2:1:3);
gotoxy(30,23);
esc_seq(REVER,WHITE);

```



```

cursor_off;
mou_flush;
gaiji_get(xx[1],yy[1],1,gaiji_dat);
gaiji_get(xx[2],yy[2],2,gaiji_dat);
get(x,y,x+23,y+23,ofs(gdata[1][0]),seg(gdata[1][0]),230);
get(x,y,x+23,y+23,ofs(gdata[2][0]),seg(gdata[2][0]),230);
iro := 7;
pass_f := 0;
last_page := 0;
end;
(*      横澤      *)
procedure henkei;      (*      変形文字の作成      *)
const
  LERR = 230;
var
  kyori : array[-23..23,-23..23] of integer;
  key : integer;
  label 999,888,777,666,555;
  begin
    gotoxy(26,18);
    writein('↑');
    for l := -23 to 23 do
      for m := -23 to 23 do
        kyori[l][m] := l*m*m;
      for i := 0 to 23 do
        for j := 0 to 23 do
          begin
            tmp1 := point(LER-30+i,UP-30+j);
            tmp2 := point(LER+i,UP-30+j);
            for k := 0 to 1000 do
              begin
                key := 0;
                if tmp2 > 0
                then
                  begin
                    for l := 0 to 23 do
                      begin
                        if (kyori[l][0] > k) then goto 888;
                        for m := 0 to 23 do
                          begin
                            if (kyori[l][m] > k) then goto 777;
                            begin
                              if ((kyori[l][m]=k) and (point(LER-30+i+1,UP-30+j
+m)>0))
                              then
                                begin
                                  key := 1;
                                  pset(LERR+90+i+(1+3) div 6,UP+200+j+(m+3) div 6,iro);
                                  pset(LERR+60+i+(1+2+3) div 6,UP+200+j+(m*2+3) div 6,iro);
                                  pset(LERR+30+i+(1+3+3) div 6,UP+200+j+(m*3+3) div 6,iro);
                                  pset(LERR+i+(1+4+3) div 6,UP+200+j+(m*4+3) div 6,iro);
                                  pset(LERR-30+i+(1+5+3) div 6,UP+200+j+(m*5+3) div 6,iro);
                                end;
                                if ((kyori[l][m]=k) and (point(LER-30+i-1,UP-30+j
+m)>0))
                                then
                                  begin
                                    key := 1;
                                    pset(LERR+90+i-(1+3) div 6,UP+200+j+(m+3) div 6,iro);
                                    pset(LERR+60+i-(1+2+3) div 6,UP+200+j+(m*2+3) div 6,iro);
                                    pset(LERR+30+i-(1+3+3) div 6,UP+200+j+(m*3+3) div 6,iro);
                                    pset(LERR+i-(1+4+3) div 6,UP+200+j+(m*4+3) div 6,iro);
                                    pset(LERR-30+i-(1+5+3) div 6,UP+200+j+(m*5+3) div 6,iro);
                                  end;
                                  if ((kyori[l][m]=k) and (point(LER-30+i+1,UP-30+j
-m)>0))
                                  then
                                    begin
                                      key := 1;
                                      pset(LERR+90+i+(1+3) div 6,UP+200+j-(m+3) div 6,iro);
                                      pset(LERR+60+i+(1+2+3) div 6,UP+200+j-(m*2+3) div 6,iro);
                                      pset(LERR+30+i+(1+3+3) div 6,UP+200+j-(m*3+3) div 6,iro);
                                      pset(LERR+i+(1+4+3) div 6,UP+200+j-(m*4+3) div 6,iro);
                                      pset(LERR-30+i+(1+5+3) div 6,UP+200+j-(m*5+3) div 6,iro);
                                    end;
                                    if ((kyori[l][m]=k) and (point(LER-30+i-1,UP-30+j
-m)>0))
                                    then
                                      begin
                                        key := 1;
                                        pset(LERR+90+i-(1+3) div 6,UP+200+j-(m+3) div 6,iro);
                                        pset(LERR+60+i-(1+2+3) div 6,UP+200+j-(m*2+3) div 6,iro);
                                        pset(LERR+30+i-(1+3+3) div 6,UP+200+j-(m*3+3) div 6,iro);
                                        pset(LERR+i-(1+4+3) div 6,UP+200+j-(m*4+3) div 6,iro);
                                        pset(LERR-30+i-(1+5+3) div 6,UP+200+j-(m*5+3) div 6,iro);
                                      end;
                                    end;
                                  end;
                                end;
                                777:
                              end;
                              888:
                            end;
                            if key > 0 then goto 999;
                          end;
                        if tmp1 > 0
                        then
                          begin
                            for l := 0 to 23 do
                              begin
                                if (kyori[l][0] > k) then goto 666;
                                for m := 0 to 23 do

```

>0))

,iro);

o);

,iro);

,iro);

);

>0))

,iro);

o);

,iro);

,iro);

);

>0))

,iro);

o);

,iro);

,iro);

);

>0))

,iro);

o);

,iro);

,iro);

);

```
begin
  if (kyori[1][m] > k) then goto 555;
  begin
    if ((kyori[1][m]=k) and (point(LER+i+1,UP-30+j+m)
      then
        begin
          key := 1;
          pset(LERR-30+i+1-(1*5+3) div 6,UP+200+j+m-(m*5+3) div 6
          pset(LERR+i+1-(1*4+3) div 6,UP+200+j+m-(m*4+3) div 6,ir
          pset(LERR+30+i+1-(1*3+3) div 6,UP+200+j+m-(m*3+3) div 6
          pset(LERR+60+i+1-(1*2+3) div 6,UP+200+j+m-(m*2+3) div 6
          pset(LERR+90+i+1-(1+3) div 6,UP+200+j+m-(m+3) div 6,iro
        end;
      if ((kyori[1][m]=k) and (point(LER+i-1,UP-30+j+m)
        then
          begin
            key := 1;
            pset(LERR-30+i-1+(1*5+3) div 6,UP+200+j+m-(m*5+3) div 6
            pset(LERR+i-1+(1*4+3) div 6,UP+200+j+m-(m*4+3) div 6,ir
            pset(LERR+30+i-1+(1*3+3) div 6,UP+200+j+m-(m*3+3) div 6
            pset(LERR+60+i-1+(1*2+3) div 6,UP+200+j+m-(m*2+3) div 6
            pset(LERR+90+i-1+(1+3) div 6,UP+200+j+m-(m+3) div 6,iro
          end;
        if ((kyori[1][m]=k) and (point(LER+i+1,UP-30+j-m)
          then
            begin
              key := 1;
              pset(LERR-30+i+1-(1*5+3) div 6,UP+200+j-m+(m*5+3) div 6
              pset(LERR+i+1-(1*4+3) div 6,UP+200+j-m+(m*4+3) div 6,ir
              pset(LERR+30+i+1-(1*3+3) div 6,UP+200+j-m+(m*3+3) div 6
              pset(LERR+60+i+1-(1*2+3) div 6,UP+200+j-m+(m*2+3) div 6
              pset(LERR+90+i+1-(1+3) div 6,UP+200+j-m+(m+3) div 6,iro
            end;
          if ((kyori[1][m]=k) and (point(LER+i-1,UP-30+j-m)
            then
              begin
                key := 1;
                pset(LERR-30+i-1+(1*5+3) div 6,UP+200+j-m+(m*5+3) div 6
                pset(LERR+i-1+(1*4+3) div 6,UP+200+j-m+(m*4+3) div 6,ir
                pset(LERR+30+i-1+(1*3+3) div 6,UP+200+j-m+(m*3+3) div 6
                pset(LERR+60+i-1+(1*2+3) div 6,UP+200+j-m+(m*2+3) div 6
                pset(LERR+90+i-1+(1+3) div 6,UP+200+j-m+(m+3) div 6,iro
              end;
            end;
          end;
          555:
          end;
          666:
          end;
          if key > 0 then goto 999;
        end;
      end;
    end;
  end;
  999:
  end;
```

```
end;
999:
end;
textcolor(2);
writeln('もういいかい 左ボタン');
textcolor(7);
mou_flush;
repeat
  mousepget(x,y,lmb,rmb);
until (lmb = 1) or (rmb = 1);
if (rmb = 1) then pass_f := 1;
```

end; (* 横澤 終わり *)

procedure conv16(byte10:byte);

const

table : string[16] = '0123456789abcdef';

var

h_byte : byte ; (* upper 4 bits of the byte *)
l_byte : byte ; (* lower 4 bits of the byte *)
upper,lower : char ;

begin

h_byte := byte10 div 16;
l_byte := byte10 mod 16;
upper := copy(table,h_byte+1,1);
lower := copy(table,l_byte+1,1);
write(upper,lower);

```

end;
procedure kan_input(hide:integer) ; (* 漢字入力 *)
type
  str80 = string[80];
var
  getsto      : string[120]      ; (* ITITARO read buffer *)
  cuf         : putbuf           ; (* jis - sjis convert buf *)
  xpos       : integer          ; (* xposition 100 + xpos*40 *)
  cx         : integer          ; (* x position of cursor *)
  s_jis      : array[0..1] of byte ; (* shidt JIS buffer *)
  xxx       : array[0..20] of str80;
  yyy       : byte              ;
begin
  cls;
  clrscr;
  atok_call(ROMA_KAN);
  atok_call(REN_HEN);
  esc_seq(REVER,WHITE);
  gotoxy(1,3);
  xxx[1] := '          かな漢字変換は一大郎を使用しています';
  xxx[2] := '          詳しい変換ルールは一大郎のマニュアルを見て下さい';
  xxx[3] := '          * * * * *          大変簡単なマニュアル          * * * * *';
  xxx[4] := '          * * * * *          1. ローマ字かな漢字変換のルールで入力して';
  xxx[5] := '          * * * * *          入力ミスしたらBSキーで';
  xxx[6] := '          * * * * *          2. スペース・キーで変換指示';
  xxx[7] := '          * * * * *          3. さらに、スペース・キーで次候補';
  xxx[8] := '          * * * * *          4. ←で文節の句切りを移動できます';
  xxx[9] := '          * * * * *          5. ↓でその文節の変換を確定します';
  xxx[10] := '          * * * * *          6. [RETURN]で入力を終了します';
  xxx[11] := '          * * * * *          *、[f.10]で入力モードを次の様に変更できます';
  xxx[12] := '          * * * * *          ローマ漢一かな漢一半角一コード一記号';
  xxx[13] := '          * * * * *          また、コード入力時には[SHIFT] [[.4]で';
  xxx[14] := '          * * * * *          J I S -区点-ソフトJ I S を変更できます';
  xxx[15] := '          * * * * *          * * * * *';
  xxx[16] := '          * * * * *          * * * * *';
  for yyy:= 01 to 16 do
  begin
    fastwrite(0,yyy,$e5,xxx[yyy]);
  end;
  gotoxy(0,17);
  esc_seq(REV_BL, GREEN);
  write('          文字を入力して下さい---->');
  esc_seq(RESET,0);
  readln(getsto);
  atok_call(END_ITI);
  x := 100;
  y := 100;
  iro := 7;
  mousexrange(0,639);
  mouseerase;
  clrscr;
  gotoxy(1,9);
  writeln('ソフトJ I S');
  gotoxy(1,10);
  writeln('J I Sコード');
  gotoxy(1,11);
  writeln('区点コード');
  for k:=1 to (integer(getsto[0]) div 2) do
  begin
    reset(fp);
    cuf[0] := integer(getsto[k*2-1]);
    cuf[1] := integer(getsto[k*2]);
    s_jis[0] := cuf[0];
    s_jis[1] := cuf[1];
    sjtojis(cuf);
    tmp1 := 90;
    tmp2 := cal_se(cuf);
    if tmp2 < 525
    then tmp1 := 30;
    seek(fp,tmp2);
    l := 0;
    repeat (* リニア・サーチ *)
      blockread(fp,buf,l,m);
      tmp2 := tmp2 + 1;
      if (cuf[0] - buf[14]) > 1
      then
        begin
          tmp2 := tmp2 + tmp1 * (cuf[0]-buf[14]-1);
          seek(fp,tmp2);
        end;
      l := l + 1;
      if l > 1000
      then
        begin
          writeln('辞書に該当するコードがありません');
          cursor_on;
          exit;
        end;
    until (buf[14] = cuf[0]) and (buf[15] = cuf[1]);
    lmb := 0;
    hide2 := tmp2 - 1;
    cx := x div 8 ;
    gotoxy(cx+1,9);
    conv16(s_jis[0]);
    conv16(s_jis[1]);
    gotoxy(cx+1,10);
    conv16(cuf[0]);
    conv16(cuf[1]);
    gotoxy(cx+1,11);
    write((buf[16]-48):1,(buf[17]-48):1,(buf[18]-48):1,(buf[19]-48):1);
    kanp24(x,y,iro,buf);
  end;
end;

```

```

        x := x + 40;
    end;
    gotoxy(3,3);
    textcolor(7);
    write('マウスの左のボタンで文字を選ぶ');
    mou_flush;
    mouseerase;
    for k:=1 to hide do
    begin
        case k of
            1:
                begin
                    gotoxy(20,12);
                    writeln('編集文字');
                end;
            2:
                begin
                    gotoxy(30,12);
                    writeln('参照文字');
                end;
        end;

        lmb := 0;
        x := 100;
        y := 100;
        while lmb < 1 do
        begin
            delay(50);
            mousepget(x,y,lmb,rmb);
            delay(100);
            xpos := ((x - 100) div 40);
            x := xpos*40 + 100;
            y := 100;
            get(x,y,x+23,y+23,ofs(gdata[k][0]),seg(gdata[k][0]),230);
            line(x,y,x+23,y+23,7,1,0,0,0,0,0);
            delay(100);
            put(x,y,ofs(gdata[k][0]),seg(gdata[k][0]),230,0,0,0,0);
            put(xx[k],yy[k],ofs(gdata[k][0]),seg(gdata[k][0]),230,0,0,0,0);
        end;
        mou_flush;
    end;
    gotoxy(1,1);
    gaiji_get(xx[1],yy[1],1,gaiji_dat);
    if hide = 2 then
        gaiji_get(xx[2],yy[2],2,gaiji_dat);
    end;
procedure cp_l2; (* パターン1 と パターン2 交換 *)
var
    gaiji_tmp : integer ;
    gdata_tmp : byte ;
begin
    for x:= 0 to 23 do
        for y := 0 to 23 do
            begin
                gaiji_tmp := gaiji_dat[1][x][y];
                gaiji_dat[1][x][y] := gaiji_dat[2][x][y];
                gaiji_dat[2][x][y] := gaiji_tmp;
            end;
        for i:= 0 to 230 do
            begin
                gdata_tmp := gdata[1][i];
                gdata[1][i] := gdata[2][i];
                gdata[2][i] := gdata_tmp;
            end;
        end;
end;
procedure mv_l2; (* パターン1 から パターン2 *)
begin
    for x:= 0 to 23 do
        for y := 0 to 23 do
            begin
                gaiji_dat[2][x][y] := gaiji_dat[1][x][y];
            end;
        for i:= 0 to 230 do
            begin
                gdata[2][i] := gdata[1][i];
            end;
        end;
end;
procedure pic_copy;
var
    wid1,wid2: integer ;
    wid : integer ;
    tmp_r1 : real ;
    tmp_r2 : real ;
    ggdata: array[0..1,0..3000] of byte ;
begin
    wid1 := 3;
    wid2 := 3;
    mouseerase;
    while (lmb = 0) or (rmb = 0) do
        begin
            mousepget(x,y,lmb,rmb);
            if rmb = 1
                then wid1 := wid1 + 2;
            if wid1 > 300
                then wid1 := 3;
            if lmb = 1
                then wid2 := wid2 + 2;
            if wid2 > 300
                then wid2 := 3;
            wid := ((wid1+8) div 8)*(wid2+1)*3+4;

```

```

if wid > 20000
then
begin
wid := ((wid1+8) div 8)*(wid2+1)*3+4;
wid2 := 5;
end;
xor_line(x,y,x+wid1,y,5,XORTT);
xor_line(x,y,x,y+wid2,5,XORTT);
xor_line(x+wid1,y,x+wid1,y+wid2,5,XORTT);
xor_line(x,y+wid2,x+wid1,y+wid2,5,XORTT);
delay(30);
xor_line(x,y,x+wid1,y,5,XORTT);
xor_line(x,y,x,y+wid2,5,XORTT);
xor_line(x+wid1,y,x+wid1,y+wid2,5,XORTT);
xor_line(x,y+wid2,x+wid1,y+wid2,5,XORTT);
end;
mou_flush;
(* make data *)
tmp_r1 := wid1 / 24.0;
tmp_r2 := wid2 / 24.0;
if (wid1>48) or (wid2>48)
then
for i:= 0 to 23 do
for j := 0 to 23 do
begin
if (point(x+trunc(tmp_r1*j),y+trunc(tmp_r2*i))>0)
or (point(x+trunc(tmp_r1*j)+1,y+trunc(tmp_r2*i))>0)
then pset(LER-30+j,UP-30+i,7)
else preset(LER-30+j,UP-30+i);
end
else
for i:= 0 to 23 do
for j := 0 to 23 do
begin
if (point(x+trunc(tmp_r1*j),y+trunc(tmp_r2*i))>0)
then pset(LER-30+j,UP-30+i,7)
else preset(LER-30+j,UP-30+i);
end;
gaiji_get(LER-30,UP-30,1,gaiji_dat);
get(LER-30,UP-30,LER-30+23,UP-30+23,ofs(gdata[1][0]),seg(gdata[1][0]),230);
writeln('');
end;
end;

```

```

procedure gazoget;

```

```

var
tmp1 : integer ; (* temporary *)
off : integer ;
iro : integer ;
data : array[0..1000] of byte ; (* get data of byte *)
da_dam : array[0..200] of byte;
i,j,k : integer ; (* loop counter or temp number *)
fp : file ;
const
KAIE = 1140;
VB = $a800;
VR = $b000;
VG = $b800;
label
OWARI;
begin
cls;
clrscr;
esc_seq(REVER,WHITE);
writeln(' ');
writeln(' 画像データを読み込む ');
writeln(' ルーチン ですよ ');
writeln(' ');
esc_seq(RESET,0);
writeln(' ');
writeln(' ');
writeln(' 画像データ入力ファイル名を入れて下さい [0で始まるファイル名:終了] ');
readln(ans1);
if ans1[0] = '0' then goto OWARI;
cls;
clrscr;
assign(fp,ans1);
reset(fp);
off := 0;
iro := 0;
for j := 0 to 380 do
begin
blockread(fp,data,1,tmp1);
{
blockread(fp,da_dam,1,tmp1);
blockread(fp,da_dam,1,tmp1);
}
for i:= 0 to 79 do
begin
mem[VR:off+i] := ( data[i+2]);
mem[VG:off+i] := ( data[i+2]);
mem[VB:off+i] := ( data[i+2]);
end;
off := off + 80;
end;
(* mouse *)
pic_copy;
close(fp);
OWARI:
cls;
end;

```



```
procedure bsort(cou:integer;var data,mata:sortbuf;ookijyun:integer);
```

```
var
  i,j,k,tmp,cou2: integer ;
begin
  if cou > 1 then
  begin
    cou2 := cou - 1;
    if cou2 < 2 then cou2 := 2;
    for i:= cou downto cou2 do
      for j:=1 to i-1 do
        begin
          if ookijyun = 0 then
            begin
              if data[j] >= data[j+1] then
                begin
                  tmp := data[j];
                  data[j] := data[j+1];
                  data[j+1] := tmp;
                  tmp := mata[j];
                  mata[j] := mata[j+1];
                  mata[j+1] := tmp;
                end;
              end;
            end;
          if ookijyun = 1 then
            begin
              if data[j] <= data[j+1] then
                begin
                  tmp := data[j];
                  data[j] := data[j+1];
                  data[j+1] := tmp;
                  tmp := mata[j];
                  mata[j] := mata[j+1];
                  mata[j+1] := tmp;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;
```

```
procedure senmitu3(var mata,m500:sortbuf);
```

```
var
  x,y : integer ;
  xx,yy : integer ;
  line1,line2: integer ;
  i,j,k,ll : integer ;
  rui_do2: integer ;
  ref_dat: getbuf ; (* char. read buf *)
  r_byte : integer ; (* real read bytes *)
  fsize : integer ; (* max file size *)
  isum : integer ;
  tmp_real : real ;
  tmp_int : integer ;
  m2wa,r2wa : real ;
  im2wa,ir2wa,rui_do,rui_min: integer;
  lpu,lpd,lp1,lpr: array[0..23] of byte ;
  lpu2,lpd2,lp12,lpr2:array[0..47] of byte;
  limit : integer ;
  fs : file ;
  osh: array[0..12] of string[10];
```

```
label
```

```
OUT1;
```

```
begin
```

```
osh[0] := '只';
osh[1] := '会';
osh[2] := 'が';
osh[3] := 'ん';
osh[4] := 'ば';
osh[5] := 'つ';
osh[6] := 'て';
osh[7] := 'お';
osh[8] := 'り';
osh[9] := 'ま';
osh[10] := 'す';
osh[11] := ' ';
assign(fs,'d:senmitu.dat');
reset(fs);
limit := 60*(ue + rue + lue + yoko);
for x:= 0 to ll do (* 上方向から *)
  begin
    xx := x * 2;
    line1 := 0;
    line2 := 0;
    for y:=0 to 23 do
      begin
        if((point(100+xx,299+y)=0) and (point(100+xx,300+y)=7))
          then
            line1 := line1 + 1;
        if((point(101+xx,299+y)=0) and (point(101+xx,300+y)=7))
          then
            line2 := line2 + 1;
        end;
        lpu[x] := (line1 + line2) * 2;
      end;
    end;
  for y:= 0 to ll do (* 左方向から *)
    begin
      yy := y * 2;
      line1 := 0;
      line2 := 0;
      for x:=0 to 23 do
        begin
```

```

        if((point(99+x,300+yy)=0) and (point(100+x,300+yy)=7))
            then
                line1 := line1 + 1;
            if((point(99+x,301+yy)=0) and (point(100+x,301+yy)=7))
                then
                    line2 := line2 + 1;
            end;
        lpl[y] := (line1 + line2)*2;
    end;
for y:= 0 to 23 do (* 左上方向から *)
begin
    j := 23 - y;
    line1 := 0;
    line2 := 0;
    for x:= 0 to j do
        begin
            if((point(99+x,299+x+y)=0) and (point(100+x,300+x+y)=7))
                then
                    line1 := line1 + 1;
            if((point(99+x+y,299+x)=0) and (point(100+x+y,300+x)=7))
                then
                    line2 := line2 + 1;
            end;
            lpl2[y] := line2;
            lpl2[y+24] := line1;
        end;
    end;
for y:= 0 to 23 do (* 右上方向から *)
begin
    j := 23 - y;
    line1 := 0;
    line2 := 0;
    for x:= 0 to j do
        begin
            if((point(124-x,299+x+y)=0) and (point(123-x,300+x+y)=7))
                then
                    line1 := line1 + 1;
            if((point(124-x-y,299+x)=0) and (point(123-x-y,300+x)=7))
                then
                    line2 := line2 + 1;
            end;
            lpr2[y] := line2;
            lpr2[y+24] := line1;
        end;
    end;
for j := 0 to 11 do
begin
    k := j*4;
    lpd[j] := lpl2[k]+lpl2[k+1]+lpl2[k+2]+lpl2[k+3];
end;
for j := 0 to 11 do
begin
    k := j*4;
    lpr[j] := lpr2[k]+lpr2[k+1]+lpr2[k+2]+lpr2[k+3];
end;
for j := 3 to 14 do
    ref_dat[j] := lpu[j-3]; (* 上 *)
for j := 15 to 26 do
    ref_dat[j] := lpd[j-15]; (* 左上 *)
for j := 27 to 38 do
    ref_dat[j] := lpl[j-27]; (* 横 *)
for j := 39 to 50 do
    ref_dat[j] := lpr[j-39]; (* 右上 *)
m2wa := 0;
for j:= 3 to 50 do
begin
    m2wa := m2wa + ref_dat[j]*ref_dat[j];
end;
m2wa := sqrt(m2wa);
seek(fs,200);
l := 0;
i := 200;
if syakudo = L1 then
begin
    rui_min := 1000;
    while (i <=3200) do
        begin
            blockread(fs,moj_dat,300,tmp1);
            for m:=0 to 299 do
                begin
                    rui_do := 0;
                    for j:= 3 to 50 do
                        begin
                            rui_do := rui_do+abs(moj_dat[m][j]-ref_dat[j]);
                        end;
                    if rui_do < rui_min then
                        begin
                            rui_min := rui_do;
                            koko := i + m;
                        end;
                    if rui_do < limit then
                        begin
                            l := l + 1;
                            m500[l] := i + m;
                            mata[l] := rui_do;
                        end;
                    if hide2 = (i + m) then
                        koko2 := rui_do;
                end;
            end;
            i := i + 300;
        end;
end;

```

```

        bsort(l,mata,m500,1);
    end;
    if syakudo = L2 then
    begin
    rui_min := 1000;
    while (i <=3200) do
    begin
        blockread(fs,moj_dat,300,tmp1);
        for m:=0 to 299 do
        begin
            rui_do := 0;
            for j:= 3 to 50 do
            begin
                rui_do2 := moj_dat[m][j]-ref_dat[j];
                rui_do := rui_do+rui_do2*rui_do2;
                if rui_do > 1000 then
                begin
                    rui_do := 10000;
                    goto OUT1;
                end;
            end;
        end;
        rui_do := round(sqrt(rui_do));

        if rui_do < rui_min then
        begin
            rui_min := rui_do;
            koko := i + m;
        end;
        if rui_do < limit then
        begin
            l := l + 1;
            m500[l] := i + m;
            mata[l] := rui_do;
        end;
        if hide2 = (i + m) then
            koko2 := rui_do;
        end;
        i := i + 300;
    end;
    bsort(l,mata,m500,1);
    end;
    if syakudo = RUI then
    begin
    rui_min := 0;
    i := 200;
    l := 0;
    ll := 0;
    gotoxy(20,17);
    write('_____');
    gotoxy(20,17);
    while (i <=3200) do
    begin
        write(osh[ll]);
        ll := ll + 1;
        blockread(fs,moj_dat,300,tmp1);
        for m:=0 to 299 do
        begin
            tmp_real := 0;
            tmp_int := 0;
            for j:= 3 to 50 do
            begin
                tmp_real := tmp_real+(moj_dat[m][j]*ref_dat[j]);
            end;
            r2wa := moj_dat[m][51]*256+moj_dat[m][52];
            if (r2wa>0) and (m2wa>0)
            then
                rui_do := trunc((tmp_real / (r2wa*m2wa))*10000);
            if rui_do > rui_min then
            begin
                rui_min := rui_do;
                koko := i + m;
            end;
            if rui_do > 900 then
            begin
                l := l + 1;
                m500[l] := i + m;
                mata[l] := rui_do;
            end;
            if hide2 = (i + m) then
                koko2 := rui_do;
            end;
            i := i + 300;
        end;
    end;
    bsort(l,mata,m500,0);
    end;
    close(fs);

    end;
procedure messyu(var mata,m500:sortbuf);
var
    x,y : integer ;
    i,j,k,ll,lll: integer ;
    ref_dat: getbuf ; (* char. read buf *)
    r_byte : integer ; (* real read bytes *)
    fsize : integer ; (* max file size *)
    isum : integer ;
    tmp_int: integer ;
    rui_do,rui_min: integer ;
    rui_do2: integer ;
    rowd :array[0..23,0..23] of byte;
    limit : integer ;

```

```

fs      : file      ;
tmp_real : real    ;
m2wa,r2wa : real   ;
osh: array[0..12] of string[10];

```

```

const
UNITSIZE = 38;

```

```

label

```

```

OUT1;
begin

```

```

osh[0] := 'A';
osh[1] := 'T';
osh[2] := 'R';
osh[3] := '初';
osh[4] := '體';
osh[5] := '算';
osh[6] := '機';
osh[7] := '構';
osh[8] := '研';
osh[9] := '究';
osh[10] := '所';
osh[11] := ' ';
assign(fs,'d:messyu.dat');
reset(fs,UNITSIZE);
tmp_int := 0;
for x:=0 to 23 do
  for y:=0 to 23 do
    begin
      rowd[x][y] := 0;
      if (point(100+x,300+y) > 0) then
        rowd[x][y] := 1;
      end;
    end;
  end;
  tmp := 0;
  for i:= 0 to 5 do
    for j:=0 to 5 do
      begin
        ref_dat[tmp] := 0;
        for k:=0 to 3 do
          for lll:=0 to 3 do
            begin
              ref_dat[tmp] := ref_dat[tmp] + rowd[j*4+lll][i*4+k];
            end;
          end;
        tmp := tmp + 1;
      end;
    end;
  end;
  m2wa := 0;
  for j:= 0 to 35 do
    begin
      m2wa := m2wa + ref_dat[j]*ref_dat[j];
    end;
  end;
  m2wa := sqrt(m2wa);
  seek(fs,200);
  l := 0;
  i := 200;
  if syakudo = L1 then
    begin
      rui_min := 1000;
      limit := 130;
      while (i <=3200) do
        begin
          blockread(fs,mo_dat,300,tmp1);
          for m:=0 to 299 do
            begin
              rui_do := 0;
              for j:= 0 to 35 do
                begin
                  rui_do := rui_do+abs(mo_dat[m][j]-ref_dat[j]);
                end;
              if rui_do < rui_min then
                begin
                  rui_min := rui_do;
                  koko := i + m;
                end;
              end;
              if rui_do < limit then
                begin
                  l := l + 1;
                  m500[l] := i + m;
                  mata[l] := rui_do;
                end;
              end;
              if hide2 = (i + m) then
                koko2 := rui_do;
            end;
          end;
          i := i + 300;
        end;
      end;
      bsort(l,mata,m500,1);
    end;
  if syakudo = L2 then
    begin
      rui_min := 1000;
      limit := 200;
      while (i <=3200) do
        begin
          blockread(fs,mo_dat,300,tmp1);
          for m:=0 to 299 do
            begin
              rui_do := 0;
              for j:= 0 to 35 do
                begin
                  rui_do2 := mo_dat[m][j]-ref_dat[j];
                  rui_do := rui_do+rui_do2*rui_do2;
                  if (rui_do>400) then
                    begin

```

```

        rui_do := 10000;
        goto OUT1;
    end;
    rui_do := round(sqrt(rui_do));
OUT1:
    if rui_do < rui_min then
    begin
        rui_min := rui_do;
        koko    := i + m;
    end;
    if rui_do < limit then
    begin
        l      := l + 1;
        m500[l] := i + m;
        mata[l] := rui_do;
    end;
    if hide2 = (i + m) then
        koko2 := rui_do;
    end;
    i := i + 300;
end;
bsort(l,mata,m500,1);
end;
if syakudo = RUI then
begin
    rui_min := 0;
    i := 200;
    l := 0;
    ll := 0;
    gotoxy(20,17);
    write('_____');
    gotoxy(20,17);
    while (i <= 3200) do
    begin
        write(osh[ll]);
        ll := ll + 1;
        blockread(fs,mo_dat,300,tmp1);
        for m:=0 to 299 do
        begin
            tmp_real := 0;
            tmp_int  := 0;
            for j:= 0 to 35 do
            begin
                tmp_real := tmp_real+(mo_dat[m][j]*ref_dat[j]);
            end;
            r2wa := mo_dat[m][36]*256+mo_dat[m][37];
            if (r2wa>0) and (m2wa>0)
            then
                rui_do := trunc((tmp_real / (r2wa*m2wa))*10000);
            if rui_do > rui_min then
            begin
                rui_min := rui_do;
                koko    := i + m;
            end;
            if rui_do > 900 then
            begin
                l := l + 1;
                m500[l] := i + m;
                mata[l] := rui_do;
            end;
            if hide2 = (i + m) then
                koko2 := rui_do;
            end;
            i := i + 300;
        end;
        bsort(l,mata,m500,0);
    end;
    close(fs);
end;
procedure
var
    moji_nin(syuhou:integer);
    x,y      : integer      ;
    i,j,k,m,ll : integer    ;
    ref_dat: getbuf        ; (* char. read buf *)

    rui_do   : integer      ;
    rui_do2  : integer      ;
    f_ruido  : real         ;
    tmp_real  : real         ;
    tmp_int   : integer      ;
    im2wa,ir2wa : integer    ;
    m2wa,r2wa : real         ;
    r_byte   : integer      ; (* real read bytes *)
    fsize    : integer      ; (* max file size *)
    oshirase : array[0..12] of string[10];
    rui_min: integer        ;
    m500    : sortbuf       ;
    mata    : sortbuf       ;
    lpu,lpd,lp1,lpr: array[0..23] of byte ;
    lpu2,lpd2,lp12,lpr2:array[0..23] of byte;
    mx,my    : integer      ;
label OUT1,OUT2;
begin
    rui_min :=10000;
    koko    := 10;
    oshirase[0] := '人';
    oshirase[1] := '生';
    oshirase[2] := 'は';
    oshirase[3] := '長';
    oshirase[4] := 'い';

```

```

oshirase[5] := 'い';
oshirase[6] := 'い';
oshirase[7] := 'い';
oshirase[8] := 'い';
oshirase[9] := 'い';
oshirase[10] := 'い';
oshirase[11] := 'い';
assign(fr, 'd:peiphe.dat');
reset(fr);
mou_flush;
mousexrange(200,344);
mouseyrange(70,200);
mousepset(270,70);
repeat
  mousepget(x,y,lmb,rmb);
  if (y<300)
    then
      begin
        mx := LER;
        my := UP-30;
      end;
  if (y<300) and (x<300)
    then
      begin
        mx := LER-30;
        my := UP-30;
      end;
  line(mx-1,my-1,mx+24,my+24,4,1,0,0,0,0,0,0);
  delay(200);
  line(mx-1,my-1,mx+24,my+24,0,1,0,0,0,0,0,0);
until (lmb = 1);
line(mx-1,my-1,mx+24,my+24,2,1,0,0,0,0,0,0);
mou_flush;
get(mx,my,mx+23,my+23,ofs(gdata[3][0]),seg(gdata[3][0]),230);
put(100,300,ofs(gdata[3][0]),seg(gdata[3][0]),230,0,0,0,0);
if (syuhou < 3) then
  begin
    for x:= 0 to 23 do (* 上方向から *)
      begin
        y := 0;
        while((point(100+x,300+y) = 0) and (y<24)) do
          y := y + 1;
        lpu[x] := y;
        while((point(100+x,300+y) = 7) and (y<24)) do
          y := y + 1;
        while((point(100+x,300+y) = 0) and (y<24)) do
          y := y + 1;
        lpu2[x] := y;
      end;
    for x:= 0 to 23 do (* 下方向から *)
      begin
        y := 23;
        while((point(100+x,300+y) = 0) and (y>=0)) do
          y := y - 1;
        lpd[x] := 23 - y;
        while((point(100+x,300+y) = 7) and (y>=0)) do
          y := y - 1;
        while((point(100+x,300+y) = 0) and (y>=0)) do
          y := y - 1;
        lpd2[x] := 23 - y;
      end;
    for y:= 0 to 23 do (* 左方向から *)
      begin
        x := 0;
        while((point(100+x,300+y) = 0) and (x<24)) do
          x := x + 1;
        lpl[y] := x;
        while((point(100+x,300+y) = 7) and (x<24)) do
          x := x + 1;
        while((point(100+x,300+y) = 0) and (x<24)) do
          x := x + 1;
        lpl2[y] := x;
      end;
    for y:= 0 to 23 do (* 右方向から *)
      begin
        x := 23;
        while((point(100+x,300+y) = 0) and (x>=0)) do
          x := x - 1;
        lpr[y] := 23 - x;
        while((point(100+x,300+y) = 7) and (x>=0)) do
          x := x - 1;
        while((point(100+x,300+y) = 0) and (x>=0)) do
          x := x - 1;
        lpr2[y] := 23 - x;
      end;
    k := 0;
    for j:= 0 to 5 do
      begin
        ref_dat[3+k] := lpu[j*4] + lpu[j*4+1] + lpu[j*4+2] + lpu[j*4+3];
        ref_dat[9+k] := lpd[j*4] + lpd[j*4+1] + lpd[j*4+2] + lpd[j*4+3];
        ref_dat[15+k] := lpl[j*4] + lpl[j*4+1] + lpl[j*4+2] + lpl[j*4+3];
        ref_dat[21+k] := lpr[j*4] + lpr[j*4+1] + lpr[j*4+2] + lpr[j*4+3];
        ref_dat[27+k] := lpu2[j*4] + lpu2[j*4+1] + lpu2[j*4+2] + lpu2[j*4+3];
        ref_dat[33+k] := lpd2[j*4] + lpd2[j*4+1] + lpd2[j*4+2] + lpd2[j*4+3];
        ref_dat[39+k] := lpl2[j*4] + lpl2[j*4+1] + lpl2[j*4+2] + lpl2[j*4+3];
        ref_dat[45+k] := lpr2[j*4] + lpr2[j*4+1] + lpr2[j*4+2] + lpr2[j*4+3];
        k := k + 1;
      end;
    m2wa := 0;
    for j:= 3 to 50 do
      begin

```

```

        m2wa := m2wa + ref_dat[j]*ref_dat[j];
    end;
    m2wa := sqrt(m2wa);
(* 文字認識1 *)

    seek(fr,200);
    l := 0;
    i := 200;

if syuhou = 2 then
    begin
    ll := 0;
    gotoxy(20,17);
    write('_____');
    gotoxy(20,17);
    if syakudo = L1 then
    begin
    while (i <=3200) do
    begin
    write(oshirase[ll]);
    ll := ll + 1;
    blockread(fr,moj_dat,300,tmp1);
    for m:=0 to 299 do
    begin
    rui_do := 0;
    for j:= 3 to 50 do
    begin
    rui_do := rui_do+abs(moj_dat[m][j]-ref_dat[j]);
    if rui_do > 600 then goto OUT1;
    end;

    if rui_do < rui_min then
    begin
    rui_min := rui_do;
    koko := i + m;
    end;
    if rui_do < 600 then
    begin
    l := l + 1;
    m500[l] := i + m;
    mata[l] := rui_do;
    end;
    if hide2 = (i + m) then
    koko2 := rui_do;
    end;
    i := i + 300;
    end;
    bsort(l,mata,m500,1);

    end;
if syakudo = L2 then
    begin
    while (i <=3200) do
    begin
    write(oshirase[ll]);
    ll := ll + 1;
    blockread(fr,moj_dat,300,tmp1);
    for m:=0 to 299 do
    begin
    rui_do2 := 0;
    for j:= 3 to 50 do
    begin
    rui_do := moj_dat[m][j]-ref_dat[j];
    rui_do2 := rui_do2+(rui_do*rui_do);
    if rui_do2 > 16900 then
    begin
    rui_do := 10000;
    goto OUT2;
    end;
    end;
    f_ruido := sqrt(rui_do2);
    rui_do :=round(f_ruido);

    if rui_do < rui_min then
    begin
    rui_min := rui_do;
    koko := i + m;
    end;
    if rui_do < 130 then
    begin
    l := l + 1;
    m500[l] := i + m;
    mata[l] := rui_do;
    end;
    if hide2 = (i + m) then
    koko2 := rui_do;
    end;
    i := i + 300;
    end;
    bsort(l,mata,m500,1);

    end;
if syakudo = RUI then
    begin
    rui_min := 0;
    i := 200;
    l := 0;
    while (i <=3200) do

```

```

begin
write(oshirase[l1]);
l1 := l1 + 1;
blockread(fr,moj_dat,300,tmp1);
for m:=0 to 299 do
begin
tmp_real := 0;
tmp_int := 0;
for j:= 3 to 50 do
begin
tmp_real := tmp_real+(moj_dat[m][j]*ref_dat[j]);
end;
r2wa := moj_dat[m][51]*256+moj_dat[m][52];
if (r2wa>0) and (m2wa>0)
then
rui_do := trunc((tmp_real / (r2wa*m2wa))*10000);
if rui_do > rui_min then
begin
rui_min := rui_do;
koko := i + m;
end;
if rui_do > 900 then
begin
l := l + 1;
m500[l] := i + m;
mata[l] := rui_do;
end;
if hide2 = (i + m) then
koko2 := rui_do;
end;
i := i + 300;
end;
bsort(l,mata,m500,0);
end;
end;
(* 線密度 *)
if syuhou = 3 then
senmitu3(mata,m500);
(* メッシュ *)
if syuhou = 4 then
messyu(mata,m500);
reset(fr);
seek(fr,koko);
blockread(fr,ref_dat,1,tmp1);
kanp24(172,300,7,ref_dat);
l1 := l - 10;
if l1 < 1 then l1 := 1;
j := 0;
for i:= 1 downto l1 do
begin
j := j + 1;
reset(fr);
seek(fr,m500[i]);
blockread(fr,ref_dat,1,tmp1);
kanp24(64+j*48,350,6,ref_dat);
gotoxy(8+j*6,22);
write(mata[i]:4);
end;
textcolor(2);
gotoxy(5,20);
writeln(' 私は を と読みます 左ボタン',koko2);
textcolor(7);
repeat
mousepget(x,y,lmb,rmb);
until (lmb = 1) or (rmb = 1);
if rmb = 1 then
begin
esc_seq(REVER,RED);
write('PLEASE ue,lue,rue,yoko ');
writeln(ue:3,lue:3,rue:3,yoko:3);
read(ue,lue,rue,yoko);
esc_seq(RESET,0);
writeln(ue:3,lue:3,rue:3,yoko:3);
delay(1000);
end;
close(fr);
end;
end;

```



```

(* 拡張グラフィックライブラリー PC-9801*)
(* 1986.12.23 KIT1 (ATR) *)
(* GDC直接駆動大変速い? *)
(* V-RAM直接書き込み大変速い *)
(* 高速 pset, preset *)
(* xor_line *)
const
  BLUE_RAM = $a800; (* V-RAM(BLUE) segment 32K *)
  RED_RAM = $b000; (* V-RAM(RED) segment 32K *)
  GREEN_RAM = $b800; (* V-RAM(GREEN) segment 32K *)
  SETT = $23;
  USETT = $22;
  XORTT = $21;
type
  str80 = string[80];
procedure scro24;
var
  i,j,k: integer ;
  begin
    for j:=283 downto 101 do
      for i:= 5 to 17 do
        begin
          k := j*80+i*2;
          memw[BLUE_RAM:k+480] := memw[BLUE_RAM:k];
          memw[RED_RAM:k+480] := memw[RED_RAM:k];
          memw[GREEN_RAM:k+480] := memw[GREEN_RAM:k];
        end;
      end;
    end;
  procedure sc24;
  var
    i,j,k: integer ;
    begin
      for j:=283 downto 101 do
        for i:= 5 to 17 do
          begin
            k := j*80+i*2;
            memw[RED_RAM:k+640] := memw[RED_RAM:k];
          end;
        end;
      end;
    procedure sc24i;
    var
      i,j,k: integer ;
      begin
        for j:=100 to 292 do
          for i:= 6 to 17 do
            begin
              k := j*80+i*2;
              memw[RED_RAM:k-1] := memw[RED_RAM:k];
            end;
          end;
        end;
      end;
    procedure sc24r;
    var
      i,j,k: integer ;
      begin
        for j:=100 to 292 do
          for i:= 16 downto 5 do
            begin
              k := j*80+i*2;
              memw[RED_RAM:k+1] := memw[RED_RAM:k];
            end;
          end;
        end;
      end;
    procedure scro24m;
    var
      i,j,k: integer ;
      begin
        for j:=101 to 285 do
          for i:= 6 to 18 do
            begin
              k := j*80+i*2;
              memw[BLUE_RAM:k] := memw[BLUE_RAM:k+480];
              memw[RED_RAM:k] := memw[RED_RAM:k+480];
              memw[GREEN_RAM:k] := memw[GREEN_RAM:k+480];
            end;
          end;
        end;
      end;
    procedure sc24m;
    var
      i,j,k: integer ;
      begin
        for j:=101 to 283 do
          for i:= 5 to 17 do
            begin
              k := j*80+i*2;
              memw[RED_RAM:k] := memw[RED_RAM:k+640];
            end;
          end;
        end;
      end;
    procedure pset_hi(x,y:integer);
    var
      i,j,p1: integer ;
      k: byte ;
      begin
        j := (x mod 8);
        k := $80 shr j;
        p1 := y*80 + (x div 8);
        mem[BLUE_RAM:p1] := (mem[BLUE_RAM:p1] or k);
        mem[RED_RAM:p1] := (mem[RED_RAM:p1] or k);
        mem[GREEN_RAM:p1] := (mem[GREEN_RAM:p1] or k);
      end;

```

```

procedure prset_hi(x,y:integer);
var
  i,j,p1: integer ;
  k      : byte   ;
begin
  j := (x mod 8);
  k := not($80 shr j);
  p1 := y*80 + (x div 8);
  mem[BLUE_RAM:p1] := (mem[BLUE_RAM:p1] and k);
  mem[RED_RAM:p1]  := (mem[RED_RAM:p1] and k);
  mem[GREEN_RAM:p1] := (mem[GREEN_RAM:p1] and k);
end;

procedure xor_line(x1,y1,x2,y2,iro:integer;kind:byte);
var
  k      : byte ;
  tmp1,tmp2 : integer ;
  addr   : integer ;
  lo_add,hi_add: byte ;
  dad    : byte ;
  dir    : byte ;
  dc,d,d2,d1,dm: integer ;
  dcl,dch : byte ;
  dl,dh   : byte ;
  d2l,d2h : byte ;
  d1l,d1h : byte ;
  dml,dmh : byte ;
begin
  repeat
    k := port[$a0] and 4;
  until (k = 4);
  (* ポート | READ | WRITE *)
  (* a 2   | エータ | コマンド *)
  (* a 0   | ステータス | パラメータ *)
  dc := x2 - x1;
  dl := y2 - y1;
  if (dc <= 0)
  then
    begin
      dc := -dc;
      if (dl < 0)
      then
        begin
          dl := -dl;
          if (dl > dc)
          then
            dir := 4;
          else
            dir := 5;
          end
        end
      else
        if (dl > dc)
        then
          dir := 7;
        else
          dir := 6;
        end
      end
    else
      begin
        if (dl <= 0)
        then
          begin
            dl := -dl;
            if (dl > dc)
            then
              dir := 3;
            else
              dir := 2;
            end
          end
        else
          if (dl > dc)
          then
            dir := 0;
          else
            dir := 1;
          end
        end;
      end;
  if (dir = 0) or (dir = 3) or (dir = 4) or (dir = 7)
  then
    begin
      tmp1 := dl;
      dl := dc;
      dc := tmp1;
    end;
  dir := dir + 8;
  dcl := lo(dc);
  dch := hi(dc);
  dl := dl*2;
  d1l := lo(dl);
  d1h := hi(dl);
  d := dl - dc;
  dl := lo(d);
  dh := hi(d);
  d2 := d - dc;
  d2l := lo(d2);
  d2h := hi(d2);
  if ((iro and 1) = 1) then
    begin
      addr := (x1 div 16) + y1 * 40;
      dad := (x1 mod 16) * 16;
      lo_addr := lo(addr);
      hi_addr := hi(addr);
    end;
end;

```

```

port[sa2]:= $78;      (* TEXTW 0111xxx xxx=0-7 レジスタ指定 *)
port[sa0]:= $ff;      (* レジスタ1 線種 文字パターン指定 *)
port[sa0]:= $ff;      (* レジスタ2 線種 と共通領域 *)
(* この場合は線種指定なので *)
(* レジスタ2~7は省略しました *)
port[sa2]:= $49;      (* CSRW 描画始点指定 *)
port[sa0]:= lo_add;   (* 下位アドレス *)
port[sa0]:= hi_add+$40; (* 上位アドレス *)
port[sa0]:= dad;      (* GDCは16ビット構成なので *)
(* 細かい指定をするには *)
(* offsetが必要です *)
port[sa2]:= $4c;      (* VECTW 描画ベクトル指定 *)
port[sa0]:= dir;      (* 描画種類 描画方向 *)
port[sa0]:= dcl;      (* delta X *)
port[sa0]:= dch;      (* 2*delta Y - delta X *)
port[sa0]:= dl;       (* *)
port[sa0]:= dh;       (* *)
port[sa0]:= d2l;      (* 2*delta Y - 2*delta X *)
port[sa0]:= d2h;      (* *)
repeat
  k := port[sa0] and 4;
until (k = 4);
port[sa0]:= dl1;      (* 2*delta Y *)
port[sa0]:= dlh;      (* XORで描きます *)
port[sa2]:= kind;     (* *)
port[sa2]:= $6c;      (* VECTE 描画始め *)
end;
if ((iro and 2) = 2) then
  begin
  addr := (xl div 16) + yl * 40;
  dad := (xl mod 16)*16;
  lo_add := lo(addr);
  hi_add := hi(addr);
  repeat
    k := port[sa0] and 4;
  until (k = 4);
  port[sa2]:= $78;      (* TEXTW 0111xxx xxx=0-7 レジスタ指定 *)
  port[sa0]:= $ff;      (* レジスタ1 線種 文字パターン指定 *)
  port[sa0]:= $ff;      (* レジスタ2 線種 と共通領域 *)
  (* この場合は線種指定なので *)
  (* レジスタ2~7は省略しました *)
  port[sa2]:= $49;      (* CSRW 描画始点指定 *)
  port[sa0]:= lo_add;   (* 下位アドレス *)
  port[sa0]:= hi_add+$80; (* 上位アドレス *)
  port[sa0]:= dad;      (* GDCは16ビット構成なので *)
  (* 細かい指定をするには *)
  (* offsetが必要です *)
  port[sa2]:= $4c;      (* VECTW 描画ベクトル指定 *)
  port[sa0]:= dir;      (* 描画種類 描画方向 *)
  port[sa0]:= dcl;      (* delta X *)
  port[sa0]:= dch;      (* 2*delta Y - delta X *)
  port[sa0]:= dl;       (* *)
  port[sa0]:= dh;       (* *)
  port[sa0]:= d2l;      (* 2*delta Y - 2*delta X *)
  port[sa0]:= d2h;      (* *)
  port[sa0]:= dl1;      (* 2*delta Y *)
  repeat
    k := port[sa0] and 4;
  until (k = 4);
  port[sa0]:= dlh;      (* XORで描きます *)
  port[sa2]:= kind;     (* *)
  port[sa2]:= $6c;      (* VECTE 描画始め *)
end;
if ((iro and 4) = 4) then
  begin
  addr := (xl div 16) + yl * 40;
  dad := (xl mod 16)*16;
  lo_add := lo(addr);
  hi_add := hi(addr);
  repeat
    k := port[sa0] and 4;
  until (k = 4);
  port[sa2]:= $78;      (* TEXTW 0111xxx xxx=0-7 レジスタ指定 *)
  port[sa0]:= $ff;      (* レジスタ1 線種 文字パターン指定 *)
  port[sa0]:= $ff;      (* レジスタ2 線種 と共通領域 *)
  (* この場合は線種指定なので *)
  (* レジスタ2~7は省略しました *)
  port[sa2]:= $49;      (* CSRW 描画始点指定 *)
  port[sa0]:= lo_add;   (* 下位アドレス *)
  port[sa0]:= hi_add+$c0; (* 上位アドレス *)
  port[sa0]:= dad;      (* GDCは16ビット構成なので *)
  (* 細かい指定をするには *)
  (* offsetが必要です *)
  port[sa2]:= $4c;      (* VECTW 描画ベクトル指定 *)
  port[sa0]:= dir;      (* 描画種類 描画方向 *)
  port[sa0]:= dcl;      (* delta X *)
  port[sa0]:= dch;      (* 2*delta Y - delta X *)
  port[sa0]:= dl;       (* *)
  port[sa0]:= dh;       (* *)
  port[sa0]:= d2l;      (* 2*delta Y - 2*delta X *)
  port[sa0]:= d2h;      (* *)
  port[sa0]:= dl1;      (* 2*delta Y *)
  repeat
    k := port[sa0] and 4;
  until (k = 4);
  port[sa0]:= dlh;      (* XORで描きます *)
  port[sa2]:= kind;     (* *)
  port[sa2]:= $6c;      (* *)
end;

```

end;

end;

procedure fastwrite(col,row,attrib : byte ;var str : str80);

begin

inline

```

($1e/
$1e/
$8a/$86/row/
$b3/$50/
$f6/$e3/
$2b/$db/
$8a/$9e/col/
$03/$c3/
$03/$c0/
$8b/$f8/
$b7/$00/
$c4/$b6/str/
$2b/$c9/
$26/$8a/$0c/
$2b/$c0/
$1f/
$22/$c9/
$74/$03/
$eb/$04/$90/
$e9/$aa/$00/
$ba/$a000/
$8c/$da/
$b6/$00/
$8a/$96/attrib/
$46/
$26/$8a/$1c/
$80/$fb/$80/
$72/$4d/
$80/$fb/$e0/
$73/$05/
$80/$fb/$a0/
$73/$43/
$fe/$c9/
$46/
$26/$8a/$3c/
$86/$fb/
$81/$fb/$2020/
$75/$03/
$bb/$2121/
$80/$ff/$80/
$72/$2d/
$80/$ff/$a0/
$73/$06/
$80/$ef/$70/
$eb/$09/$90/
$80/$ff/$f0/
$73/$1d/
$80/$ef/$b0/
$0a/$db/
$79/$02/
$fe/$cb/
$02/$ff/
$80/$fb/$9e/
$72/$06/
$80/$eb/$5e/
$eb/$03/$90/
$fe/$cf/
$80/$eb/$1f/
$86/$fb/
$e4/$60/
$a8/$04/
$74/$fa/
$b0/$0d/
$e6/$62/
$fa/
$e4/$60/
$a8/$04/
$74/$ef/
$80/$ff/$00/
$74/$03/
$80/$eb/$20/
$89/$1d/
$80/$ff/$00/
$74/$09/
$47/
$47/
$80/$c7/$80/
$89/$1d/
$4f/
$4f/
$1e/
$52/
$ba/$a200/
$8e/$da/
$5a/
$89/$15/
$80/$ff/$00/
$74/$04/
$47/
$47/
$89/$15/
$1f/
$47/
```

\$47/
\$b7/\$00/
\$e2/\$03/
\$eb/\$04/\$90/
\$e9/\$61/\$ff/
\$1f):

end;

```

*      Hard copy library
  for PC-PR201T and its compatibles
    made in KITILAND
  28/Nov./1986 version 1.0
  Copy right (c) ATR 1986 *)

```

```

procedure hard_copy_screen;
const
  (* 画面のハードコピーをします *)
  (* ただし640*400ドット *)
  (* ようするにscreen(3,0,0,1)の時 *)
  (* 画面1毎あたり平均2分かかる *)
  BLUE_RAM = $a800; (* V-RAM(BLUE) segment 32K *)
  RED_RAM  = $b000; (* V-RAM(RED) segment 32K *)
  GREEN_RAM = $b800; (* V-RAM(GREEN) segment 32K *)

var
  i,j,k,l,m      : integer      ; (* counter *)
  pl             : integer      ; (* temporary *)
  tmp           : array[0..2,0..7] of byte; (* out data *)
  t1,t2,t3,t4,t5,t6,t7,t8 : byte ; (* in data *)

procedure out_printer(c : byte );
begin
  while (port[$42] and $04) <> $04 do;
    port[$40] := c; port[$46] := 14; port[$46] := 15;
  end;

begin
  write(LST,Chr($1b),Chr($4b)); (* 印字方式 漢字モード *)
  write(LST,Chr($1b),Chr($4d)); (* ドット対応グラフィック イタイプ *)
  write(LST,chr($1c),chr($42)); (* 漢字文字幅 1/5 *)
  write(LST,chr($1b),chr($65),'22'); (* 文字の拡大 4倍角 *)
  write(LST,chr($1b),chr($54),'18'); (* 改行幅の設定 18/120 *)

  (* 送出するデータの順序とV RAMの対応 *)
  (* 1column 2column 3column ... 640column *)
  LSB 1 1 4byte . . 1910byte
  . 1byte . . . .
  MSB 8 8 . . .
  9 9 . . .
  . 2byte . 5byte . . 1919byte
  16 16 . . .
  17 17 . . .
  . 3byte . 6byte . . 1920byte
  24 24 . . . *)

  for i:=0 to 16 do
    begin
      l := i * 1920;
      write(LST,Chr($1b),Chr($4a),'0640');
      (* 24ビット対応モードで640列データを送りますよ *)
      for j:=0 to 79 do
        begin
          for m:=0 to 2 do
            begin
              pl := l + j + m*640;
              t1 := mem[BLUE_RAM:pl+ 0] or mem[RED_RAM:pl+ 0] or mem[GREEN_RAM:pl+ 0];
              t2 := mem[BLUE_RAM:pl+ 80] or mem[RED_RAM:pl+ 80] or mem[GREEN_RAM:pl+ 80];
              t3 := mem[BLUE_RAM:pl+160] or mem[RED_RAM:pl+160] or mem[GREEN_RAM:pl+160];
              t4 := mem[BLUE_RAM:pl+240] or mem[RED_RAM:pl+240] or mem[GREEN_RAM:pl+240];
              t5 := mem[BLUE_RAM:pl+320] or mem[RED_RAM:pl+320] or mem[GREEN_RAM:pl+320];
              t6 := mem[BLUE_RAM:pl+400] or mem[RED_RAM:pl+400] or mem[GREEN_RAM:pl+400];
              t7 := mem[BLUE_RAM:pl+480] or mem[RED_RAM:pl+480] or mem[GREEN_RAM:pl+480];
              t8 := mem[BLUE_RAM:pl+560] or mem[RED_RAM:pl+560] or mem[GREEN_RAM:pl+560];
              for k:=0 to 7 do
                begin
                  tmp[m][k] := (t1 and 1)*1+(t2 and 1)*2+(t3 and 1)*4+(t4 and 1)*8+(t5 and 1)*16;
                  tmp[m][k] := tmp[m][k] +(t6 and 1)*32+(t7 and 1)*64+(t8 and 1)*128;
                  t1 := (t1 shr 1);
                  t2 := (t2 shr 1);

                  t3 := (t3 shr 1);
                  t4 := (t4 shr 1);
                  t5 := (t5 shr 1);
                  t6 := (t6 shr 1);
                  t7 := (t7 shr 1);
                  t8 := (t8 shr 1);
                end;
              end;
            end;
          for k:=7 downto 0 do (* プリンターへ出力 8列分 (24バイト) *)
            begin
              for m:=0 to 2 do
                begin
                  out_printer(tmp[m][k]);
                end;
              end;
            end;
          end;
        end;
      end;
      writeln(LST,Chr($0d)); (* 640ドット毎に [CR] を送る *)
      end;
      writeln(LST,chr($1b),chr($61));
      writeln(LST,Chr($1b),Chr($63),chr($31)); (* プリンタリセット *)
    end;
  end;

```

```

{ *****
  Mouse library for Turbo Pascal
  This file must be included by I compiler directive.
  This is for mouse.sys driver.
  Coded by Hideki Kawahara. 5/4/1986.
  Rev. by Hideki Kawahara. 21/4/1986.

  Cf. "Detailed description of PC-98 Extended
        System Calls", Interface separate vol., 1986-4.
  ***** }

```

```

type
  register = record
    ax,bx,cx,dx,bp,si,di,ds,es,flags : integer ;
  end ;
  figure = array[0..63] of byte;

var
  reg : register ;

procedure mouseinit;
begin
  with reg do begin
    ax := 0;
  end;
  intr($33,reg);
end;

procedure mousedisp;
begin
  with reg do begin
    ax := 01;
  end;
  intr($33,reg);
end;

procedure mouseerase;
begin
  with reg do begin
    ax := $02;
  end;
  intr($33,reg);
end;

procedure mousepget(var x,y,lmb,rmb :integer);
begin
  with reg do begin
    ax := $03;
  end;
  intr($33,reg);
  with reg do begin
    x := cx;
    y := dx;
    if ax <> 0 then
      lmb := 1
    else
      lmb := 0;
    if bx <> 0 then
      rmb := 1
    else
      rmb := 0;
  end;
end;

procedure mousepset(x,y : integer ( mouse coordinate ));
begin
  with reg do begin
    ax := $04;
    cx := x;
    dx := y;
  end;
  intr($33,reg);
end;

procedure mouselbp(var st,n,x,y : integer ( stat and number of press ));
begin
  with reg do begin
    ax := $05;
  end;
  intr($33,reg);
  with reg do begin
    st := ax;
    n := bx;
    x := cx;
    y := dx;
  end;
end;

procedure mouselbo(var st,n,x,y : integer );
begin
  with reg do begin
    ax := $06;
  end;
  intr($33,reg);
  with reg do begin
    st := ax;
    n := bx;
    x := cx;
    y := dx;
  end;
end;

```

```

procedure mouserbp(var st,n,x,y : integer );
begin
  with reg do begin
    ax := $07;
  end;
  intr($33,reg);
  with reg do begin
    st := ax;
    n := bx;
    x := cx;
    y := dx;
  end;
end;

procedure mouserbo(var st,n,x,y : integer );
begin
  with reg do begin
    ax := $08;
  end;
  intr($33,reg);
  with reg do begin
    st := ax;
    n := bx;
    x := cx;
    y := dx;
  end;
end;

procedure mouseshape(x,y          : integer ; ( center of cursor )
                    var shape : figure );
begin
  with reg do begin
    ax := $09;
    bx := x;
    cx := y;
    es := Seg(shape);
    dx := Ofs(shape);
  end;
  intr($33,reg);
end;

procedure mousedistance(var x,y : integer );
begin
  reg.ax := $0b;
  intr($33,reg);
  x := reg.cx;
  y := reg.dx;
end;

procedure mousesens(xmick,ymick : integer );
begin
  reg.ax := $0f;
  reg.cx := xmick;
  reg.dx := ymick;
  intr($33,reg);
end;

procedure mousexrange(x1,x2 : integer );
begin
  reg.ax := $10;
  reg.cx := x1;
  reg.dx := x2;
  intr($33,reg);
end;

procedure mouseyrange(y1,y2 : integer );
begin
  reg.ax := $11;
  reg.cx := y1;
  reg.dx := y2;
  intr($33,reg);
end;

procedure mousecolor(ccode : integer );
begin
  reg.ax := $12;
  reg.bx := ccode;
  intr($33,reg);
end;

```



```

(* エスケープ・シーケンス テスト *)
const
  REV_IND = 0;
  IND     = 1;
  VERTI   = 2;
  U_LINE  = 4;
  BLINK   = 5;
  REV_BL  = 6;
  REVER   = 7;
  RESET   = 8;
  BLACK   = 30;
  BLUE    = 34;
  RED     = 31;
  PURPLE  = 35;
  GREEN   = 32;
  SKYBLUE = 36;
  YELLOW  = 33;
  WHITE   = 37;

var
  loop: integer ; (* loop counter -tmp- *)

procedure esc_seq(ps1,ps2:integer);
begin
  case ps1 of
    REV_IND:
      begin
        (* Reverse Index *)
        for loop:= 1 to ps2 do
          write('');
        end;
      end;
    IND:
      begin
        (* Index *)
        for loop:= 1 to ps2 do
          write('');
        end;
      end;
    VERTI:
      begin
        (* vertical line *)
        write('',2,':m');
      end;
    U_LINE:
      begin
        (* under line *)
        write('',4,':',ps2,':m');
      end;
    BLINK:
      begin
        (* blink line *)
        write('',5,':',ps2,':m');
      end;
    REV_BL:
      begin
        (* blink line *)
        write('',5,':',ps2+10,':m');
      end;
    REVER:
      begin
        (* reverse line *)
        write('',7,':',ps2,':m');
      end;
    RESET:
      begin
        (* reset *)
        write('m');
      end;
  end;
end;

```

```
procedure err_hand; (* エラー・ハンドリング *)
begin
  (* errorptr := ofs(err_hand) *)
  if hi(err_num) = 1 then
    begin
      textcolor(6);
      writeln(' ');
      writeln(' 指定されたファイルが存在しません ');
      writeln(' ');
      textcolor(7);
    end;
  if hi(err_num) = 0 then
    writeln(' C t r l - c を押しましたね ');
  if hi(err_num) = 2 then
    begin
      textcolor(2);
      writeln(' バグ情報は内線619まで!! ');
      writeln(' PC=XXXXをメモしてね ');
      textcolor(7);
    end;
  cursor_on;
end;
```

```

( this file is JIS6234(24*24) display library )
( JIS6232(16*16) )
( avail graphic routine made in KIT1 )
( 1986.10.14 )
type
getbuf = array [0..127] of byte;
putbuf = array [0..1] of byte;
procedure jistosj(var cuf :putbuf);(* JISコードをシフトJISへ *)
begin
(* 引き数 cuf [0] JISコード上位 *)
(* cuf [1] JISコード下位 *)
(* リターン cuf [0] SJISコード上位 *)
(* cuf [1] SJISコード下位 *)
if odd(cuf[0]) = false
then
begin
cuf[1] := cuf[1] + 126;
if cuf[0] < 95
then
cuf[0] := (cuf[0] div 2) + 112
else
cuf[0] := (cuf[0] div 2) + 176;
end
else
begin
cuf[1] := cuf[1] + 31;
if cuf[0] < 95
then
cuf[0] := ((cuf[0] + 1) div 2) + 112
else
cuf[0] := ((cuf[0] + 1) div 2) + 176;
if cuf[1] >= 126
then
cuf[1] := cuf[1] + 1;
end;
end;
end;

procedure sjtojis(var cuf :putbuf);(* JISコードをシフトJISへ *)
(* 引き数 cuf [0] JISコード上位 *)
(* cuf [1] JISコード下位 *)
(* リターン cuf [0] SJISコード上位 *)
(* cuf [1] SJISコード下位 *)
begin
if cuf[1] >= 159
then
begin
cuf[1] := cuf[1] - 126;
if cuf[0] >= 224
then
cuf[0] := (cuf[0] - 176) * 2
else
cuf[0] := (cuf[0] - 112) * 2;
end
else
begin
cuf[1] := cuf[1] - 31;
if cuf[1] >= 97
then
cuf[1] := cuf[1] - 1;
if cuf[0] >= 224
then
cuf[0] := (cuf[0] - 176) * 2 - 1
else
cuf[0] := (cuf[0] - 112) * 2 - 1;
end;
end;
end;

procedure kanp16(xs,ys:integer;buf:getbuf);(* 16ドット表示 *)
(* 引き数 x, y ポジション, ドットパターンバッファ *)
(* bufは128バイト構成 *)
(* 54バイト目から表示データが存在する事 *)
(* 85バイト目に *)
var
i,j,tmp : integer ;
x,y : integer ;
begin
x := 0;
y := 0;
for j := 54 to 85 do
begin
for i := 7 downto 0 do
begin
x := x + 1;
tmp := (buf[j] shr i);
if odd(tmp) = true
then pset (xs+x,ys+y,7);
{ else preset (xs+x,ys+y); }
end;
if (j mod 2) = 1
then y := y + 1;
if (x mod 16) = 0
then x := 0;
end;
end;
end;

procedure kanp24(xs,ys,iro:integer;buf:getbuf);(* 24ドット表示 *)
(* 引き数 x, y ポジション, ドットパターンバッファ *)
(* bufは128バイト構成 *)
(* 54バイト目から表示データが存在する事 *)
(* 125バイト目に *)
var
i,j,tmp : integer ;
x,y : integer ;
begin
x := 0;

```

```

y := 0;
xs := xs - 1;
for j := 54 to 125 do
begin
for i := 7 downto 0 do
begin
x := x + 1;
tmp := (buf[j] shr i);
if odd(tmp) = true
then
pset_hi(xs+x,ys+y);
( pset (xs+x,ys+y,iro); )
( else preset (xs+x,ys+y); )
end;
if (j mod 3) = 2
then y := y + 1;
if (x mod 24) = 0
then x := 0;
end;
end;
end;
function cal_se(cuf:putbuf):integer;(* J I Sコードから辞書位置を決定する *)
begin
if (cuf[0] < 37)
then cal_se := 1
else if (cuf[0] >= 37) and (cuf[0] < 48)
then cal_se := 200
else if (cuf[0] >= 48) and (cuf[0] < 55)
then cal_se := 525
else if (cuf[0] >= 55) and (cuf[0] < 60)
then cal_se := 1100
else if (cuf[0] >= 60) and (cuf[0] < 65)
then cal_se := 1600
else if (cuf[0] >= 65) and (cuf[0] < 70)
then cal_se := 2100
else if (cuf[0] >= 70) and (cuf[0] < 75)
then cal_se := 2500
else if (cuf[0] >= 75) and (cuf[0] < 80)
then cal_se := 3000
else if (cuf[0] >= 80) and (cuf[0] < 85)
then cal_se := 3400
else if (cuf[0] >= 85) and (cuf[0] < 90)
then cal_se := 3900
else if (cuf[0] >= 90) and (cuf[0] < 100)
then cal_se := 4400
else if (cuf[0] >= 100) and (cuf[0] < 110)
then cal_se := 5300
else if (cuf[0] >= 110) and (cuf[0] < 120)
then cal_se := 6300;
end;

```

```
( *****
Graphic Library of Turbo Pascal using EGR98:
This file must be included by l compiler directive.
THIS IS FOR NEC PC9801F/E GRAPHIC LIO.
***** )
```

(C) 1984 Y.FUKADA

```
1st version .... Sept. 28, 1984
2nd version .... Oct. 1, 1984
3rd version .... Oct. 6, 1984
4th and final v. Oct. 8, 1984
added cursor control.... Oct.11
```

*****)

```
type scr = record
    arg1,arg2,arg3,arg4 : byte
end;

vi = record
    arg1,arg2,arg3,arg4 : integer;
    arg5,arg6           : byte
end;

col = record
    arg1,arg2,arg3      : byte
end;

col2= record
    arg1,arg2           : byte
end;

ps = record
    arg1,arg2           : integer;
    arg3                : byte
end;

li = record
    arg1,arg2,arg3,arg4 : integer;
    arg5,arg6,arg7,arg8,arg9,arg10 : byte;
    arg11,arg12         : integer
end;

cir = record
    arg1,arg2,arg3,arg4 : integer;
    arg5,arg6           : byte;
    arg7,arg8,arg9,arg10: integer;
    arg11               : byte;
    arg12,arg13         : integer
end;

pa = record
    arg1,arg2           : integer;
    arg3,arg4           : byte
end;

pa2 = record
    arg1,arg2           : integer;
    dummy,arg3          : byte;
    arg4,arg5           : integer;
    arg6                : byte
end;

ge = record
    arg1,arg2,arg3,arg4,arg5,arg6,arg7 : integer
end;

pu = record
    arg1,arg2,arg3,arg4,arg5 : integer;
    arg6,arg7,arg8,arg9 : byte
end;

ka = record
    arg1,arg2,arg3      : integer;
    arg4,arg5,arg6,arg7 : byte
end;

rol = record
    arg1,arg2           : integer;
    arg3                : byte
end;

po = record
    arg1,arg2           : integer
end;

pr = record
    arg1,arg2           : integer;
    arg3                : byte
end;
```

```
procedure graph(func, segbase, offset: integer); external 'PASGRAPH';
```

```
procedure init;
begin
    graph(0,0,0)
end;
```

```

procedure screen(mode,sw,ac,disp: byte);
var screen_arg: scr;
    seg_arg, offset_arg: integer;
begin
    with screen_arg do
        begin
            arg1 := mode;
            arg2 := sw;
            arg3 := ac;
            arg4 := disp
        end;
        seg_arg := seg(screen_arg);
        offset_arg := ofs(screen_arg);
        graph(1, seg_arg, offset_arg)
    end;
end;

procedure view(x1,y1,x2,y2: integer; ca,cb: byte);
var view_arg: vi;
    seg_arg, offset_arg: integer;
begin
    with view_arg do
        begin
            arg1 := x1;
            arg2 := y1;
            arg3 := x2;
            arg4 := y2;
            arg5 := ca;
            arg6 := cb
        end;
        seg_arg := seg(view_arg);
        offset_arg := ofs(view_arg);
        graph(2, seg_arg, offset_arg)
    end;
end;

procedure color(cbg,cbd,cfg: byte);
var color_arg: col;
    seg_arg, offset_arg: integer;
begin
    with color_arg do
        begin
            arg1 := cbg;
            arg2 := cbd;
            arg3 := cfg
        end;
        seg_arg := seg(color_arg);
        offset_arg := ofs(color_arg);
        graph(3, seg_arg, offset_arg)
    end;
end;

procedure color2(pal,cc: byte);
var color2_arg: col2;
    seg_arg, offset_arg: integer;
begin
    with color2_arg do
        begin
            arg1 := pal;
            arg2 := cc
        end;
        seg_arg := seg(color2_arg);
        offset_arg := ofs(color2_arg);
        graph(4, seg_arg, offset_arg)
    end;
end;

procedure cls;
begin
    graph(5,0,0)
end;

procedure pset(x,y: integer; pal: byte);
var
    pset_arg: ps;
    seg_arg, offset_arg: integer;
begin
    with pset_arg do
        begin
            arg1 := x;
            arg2 := y;
            arg3 := pal
        end;
        seg_arg := seg(pset_arg);
        offset_arg := ofs(pset_arg);
        graph(6, seg_arg, offset_arg)
    end;
end;

procedure line(x1,y1,x2,y2: integer; pall,dc,lsw,pal2: byte;
    lstyle: integer; lentile: byte; ofstile,segtile: integer);
var
    line_arg: li;
    seg_arg, offset_arg : integer;
begin
    with line_arg do
        begin
            arg1 := x1;
            arg2 := y1;
            arg3 := x2;
            arg4 := y2;
            arg5 := pall;
            arg6 := dc;
            arg7 := lsw;
            case arg6*10+arg7 of
                1,2,11,12: begin arg8 := lo(lstyle); arg9 := hi(lstyle) end;
            end;
        end;
    end;
end;

```

```

    21: arg8 := pal2;
    22: begin arg10 := lentile; arg11 := ofstile; arg12 := segtile end;
end;
end;
seg_arg := seg(line_arg);
offset_arg := ofs(line_arg);
graph(7,seg_arg,offset_arg)
end;

procedure circle(cx,cy,rx,ry: integer; pall,flag: byte;
                sx,sy,ex,ey: integer; lentile: byte;
                ofstile,segtile: integer);
var circle_arg: cir;
    seg_arg, offset_arg: integer;
begin
    with circle_arg do
        begin
            arg1 := cx;
            arg2 := cy;
            arg3 := rx;
            arg4 := ry;
            arg5 := pall;
            arg6 := flag;
            arg7 := sx;
            arg8 := sy;
            arg9 := ex;
            arg10 := ey;
            arg11 := lentile;
            arg12 := ofstile;
            arg13 := segtile
        end;
        seg_arg := seg(circle_arg);
        offset_arg := ofs(circle_arg);
        graph(8, seg_arg, offset_arg)
    end;
end;

procedure paint(x,y: integer; pall,pal2: byte);
var paint_arg: pa;
    seg_arg, offset_arg: integer;
begin
    with paint_arg do
        begin
            arg1 := x;
            arg2 := y;
            arg3 := pall;
            arg4 := pal2
        end;
        seg_arg := seg(paint_arg);
        offset_arg := ofs(paint_arg);
        graph(9, seg_arg, offset_arg)
    end;
end;

procedure paint2(x,y: integer; lentile: byte; ofstile,segtile: integer;
                 cbd: byte);
var paint2_arg: pa2;
    seg_arg,offset_arg: integer;
begin
    with paint2_arg do
        begin
            arg1 := x;
            arg2 := y;
            arg3 := lentile;
            arg4 := ofstile;
            arg5 := segtile;
            arg6 := cbd
        end;
        seg_arg := seg(paint2_arg);
        offset_arg := ofs(paint2_arg);
        graph(10, seg_arg, offset_arg)
    end;
end;

procedure get(x1,y1,x2,y2,ofs_buff,seg_buff,len: integer);
var get_arg: ge;
    seg_arg, offset_arg: integer;
begin
    with get_arg do
        begin
            arg1 := x1;
            arg2 := y1;
            arg3 := x2;
            arg4 := y2;
            arg5 := ofs_buff;
            arg6 := seg_buff;
            arg7 := len
        end;
        seg_arg := seg(get_arg);
        offset_arg := ofs(get_arg);
        graph(11, seg_arg, offset_arg)
    end;
end;

procedure put(x,y,ofs_buff,seg_buff,len: integer; dm,csw,fc,bc: byte);
var put_arg: pu;
    seg_arg, offset_arg: integer;
begin
    with put_arg do
        begin
            arg1 := x;
            arg2 := y;
            arg3 := ofs_buff;
            arg4 := seg_buff;
            arg5 := len;

```

```

    arg6 := dm;
    arg7 := csw;
    arg8 := fc;
    arg9 := bc
end;
seg_arg := seg(put_arg);
offset_arg := ofs(put_arg);
graph(12, seg_arg, offset_arg)
end;

procedure kanji(x,y,jap: integer; dm,csw,fc,bc: byte);
var kanji_arg: ka;
    seg_arg, offset_arg: integer;
begin
    with kanji_arg do
        begin
            arg1 := x;
            arg2 := y;
            arg3 := jap;
            arg4 := dm;
            arg5 := csw;
            arg6 := fc;
            arg7 := bc
        end;
        seg_arg := seg(kanji_arg);
        offset_arg := ofs(kanji_arg);
        graph(13, seg_arg, offset_arg)
    end;

procedure roll(updot,leftdot: integer; clrflag: byte);
var roll_arg: rol;
    seg_arg, offset_arg: integer;
begin
    with roll_arg do
        begin
            arg1 := updot;
            arg2 := leftdot;
            arg3 := clrflag
        end;
        seg_arg := seg(roll_arg);
        offset_arg := ofs(roll_arg);
        graph(14, seg_arg, offset_arg)
    end;

function point(x,y: integer): integer;
var
    point_arg: po;
    seg_arg, offset_arg: integer;

function ppoint(func,p1,p2: integer): integer; external 'pasgraph';

begin
    with point_arg do
        begin
            arg1 := x;
            arg2 := y;
        end;
        seg_arg := seg(point_arg);
        offset_arg := ofs(point_arg);
        point := ppoint(15,seg_arg,offset_arg)
    end;

procedure copy(option: byte);
var seg_arg, offset_arg: integer;
begin
    seg_arg := seg(option);
    offset_arg := ofs(option);
    graph(16,seg_arg,offset_arg)
end;

procedure preset(x,y: integer);
var preset_arg: pr;
    seg_arg, offset_arg: integer;
begin
    with preset_arg do
        begin
            arg1 := x;
            arg2 := y;
            arg3 := $ff
        end;
        seg_arg := seg(preset_arg);
        offset_arg := ofs(preset_arg);
        graph(17,seg_arg,offset_arg)
    end;

procedure textcolor(col:byte);
var c : string[3];
begin
    case col of
        0: c := '30m';
        1: c := '18m';
        2: c := '17m';
        3: c := '19m';
        4: c := '20m';
        5: c := '22m';
        6: c := '21m';
        7: c := '23m';
        8: c := '40m';
        9: c := '44m';
        10: c := '41m';
        11: c := '45m';

```



```
12: c := '42m';
13: c := '46m';
14: c := '43m';
15: c := '47m'
end;
write(#$1B, '[' , c)
end;

procedure use_all_screen;
begin
write(#$1B, '[>1h')
end;

procedure reset_screen;
begin
write(#$1B, '[>1l')
end;

procedure cursor_on;
begin
write(#$1B, '[>5l')
end;

procedure cursor_off;
begin
write(#$1B, '[>5h')
end;

( *****
END OF GRAPH.LIB
***** )
```