

# ロト7 高速選択ソフトウェア Kopic

ロトが抽せんされる前に、電子天秤でボールの重量が測定されます。夢ロトくんは、14グラムのボールを抽出します。過去に 14.09g(許容誤差 ± 1%)であると公表されました。Kopic は、CSV ファイルにしたがって、ボールに重みを付与できます。重いほど抽出されやすい。

## Java を実行

Java を実行できる環境が必要です。Java をインストールしてください。Kopic を利用するには、コマンドプロンプトに下記のようなコマンドを入力して、Enter を押してください。

## java Kopic

### XBM から CSV を作成

CSV は、テキストファイルであり、テキストエディタで閲覧および編集が可能です。Kopic は、XBM から CSV を作成できます。下記のコマンドは、kopic.xbm から kopic.csv を作成します。[38](#) が書き出されます。

## java Kopic c

### 引き算

XBM から CSV を作成するとき、各数値から 2 を引いて、結果を kopic.csv に書き出すには、manage.txt というテキストファイルの第 1 行に 2 を記入してください。さもないければ、コマンドプロンプトに下記のコマンドを入力して、Enter を押してください。画像を 2 ピクセルだけ左へ移動したかのように CSV が作成されます。

## java Kopic -cd2

### manage.txt

manage.txt を開くことができない場合、Kopic は、自動的に作成して第 1 行に 0 を記入します。

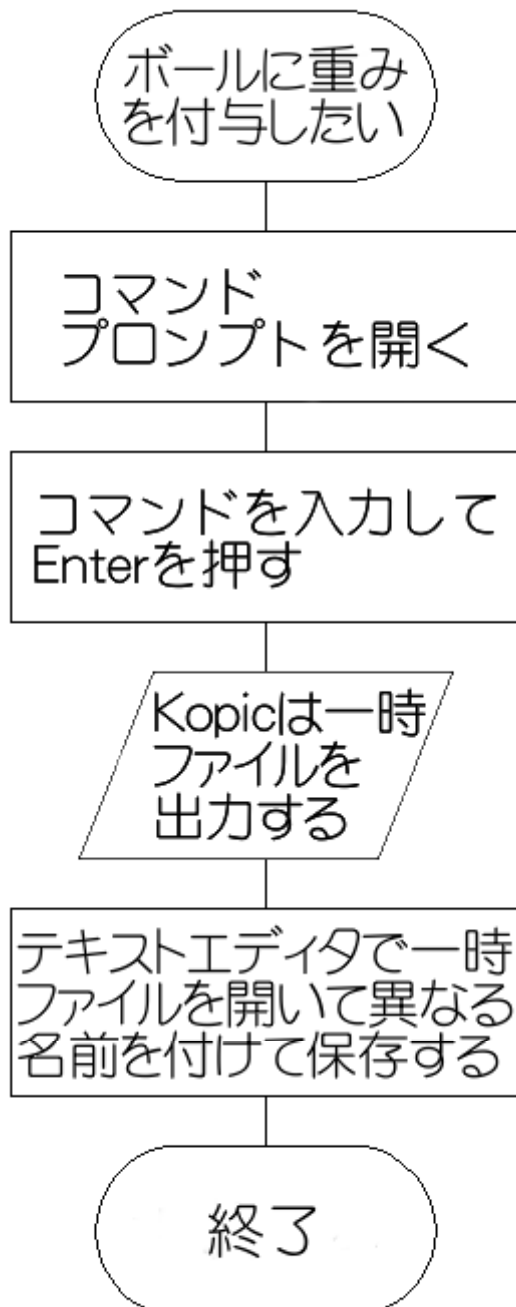
## 概要

高速選択を行うには、まずテキストエディタで kopic.csv を開いて、設定を確認してください。たとえば、Setting を 0 に設定すれば、年月日時分秒により、実行するたびに異なる結果になります。次に、コマンドプロンプトにコマンドを入力して、Enter を押してください。Kopic は、一時ファイルを出力して、コマンドプロンプトに一時ファイルの名前

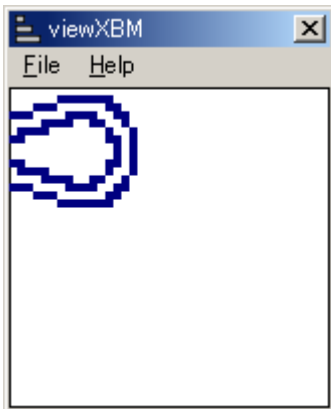
を表示します。pic 番号-分-秒.txt の形式で一時ファイルを命名します。一時ファイルと同一の内容をログに追加します。

### コマンドプロンプト

コマンドプロンプトを開くには、まず、Windows キー ( 田 ) を押しながら、R を押してください。次に、cmd を入力して、Enter を押してください。







各点として、正方形は、4の幅、4の高さで表示されています。すなわち、この表示は、真のサイズより4倍大きい。

## インストール

1. ロトサマリーというウェブサイトから KopicJava.zip をダウンロードしてください。
2. KopicJava.zip を展開してください。
3. たとえば、Cドライブに KopicJava フォルダを移動してください。
4. フォルダの名前 (C:¥KopicJava¥info¥lotosummary¥software) を変更しないでください。

## プログラムの削除

アンインストール(プログラムの削除)を行うには、フォルダ (KopicJava) ごと削除します。

## 著作権

作者として、Sogaya(そがや)は、Kopic の著作権を保有しています。ソースコードを添付しました。

## JDK

JDK 1.3.1 でも構築できます。JDK 1.8.0 で構築された Kopic の名称は、Kopic Java  
Kopic Java Copyright © 2017, Sogaya

## 免責条項

Kopic のご利用によって発生するいかなる損害も、Sogaya は、責任を負わないものとします。

## Kopic ユーザーズガイド第3版

2017年3月14日(火)。Revision 3.05904

# ランレングス符号化

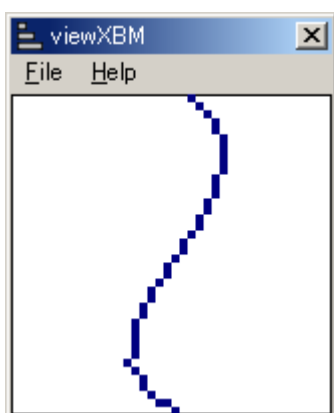
Kopic は、XBM を読み込んで、画像のランレングス符号化を行って、kopic.kun を作成

できます。kopic.xbm は、テキストファイルですが、kopic.kun は、バイナリファイルです。コマンドプロンプトに下記のコマンドを入力して、Enter を押してください。

## java Kopic k

### 圧縮

Kopic が利用するために、kopic.xbm は、左に白い領域があります。Kopic は、左から右へ白い領域がどこまで到達しているか判定して、kopic.csv を作成します。ロト7の場合、40×40のXBMとして、高さ方向へ1から37まで**切れ目がない画像**が適しています。添付のviewXBMがkopic.xbmを描画した例



こうした単調な画像であるため、画像のランレングス符号化により、ファイルのサイズが圧縮される可能性が高い。この画像になる kopic.xbm および kopic.kun のサイズを比較します。

| ファイル名     | サイズ      | 用途        |
|-----------|----------|-----------|
| kopic.xbm | 1159 バイト | 描画、CSV 作成 |
| kopic.kun | 209 バイト  | 保存        |

### 伸長

Kopic は、kopic.kun を読み込んで、画像を復元できます。まず、kopic.kun から kopic.txt を作成して、次に kopic.txt から kopic.xbm を作成してください。kopic.txt は、画像を概観できるテキストファイルです。空白は、白い点に対応しており、シャープ、ナンバーサイン、ハッシュ( # )は、黒い点に対応しています。kopic.txt を作成するには、コマンドプロンプトに下記のコマンドを入力して、Enter を押してください。

## java Kopic kopic.kun

### 真偽

白い点に対応させたい文字を false.txt の各行に記入してください。Kopic は、**1 バイト**

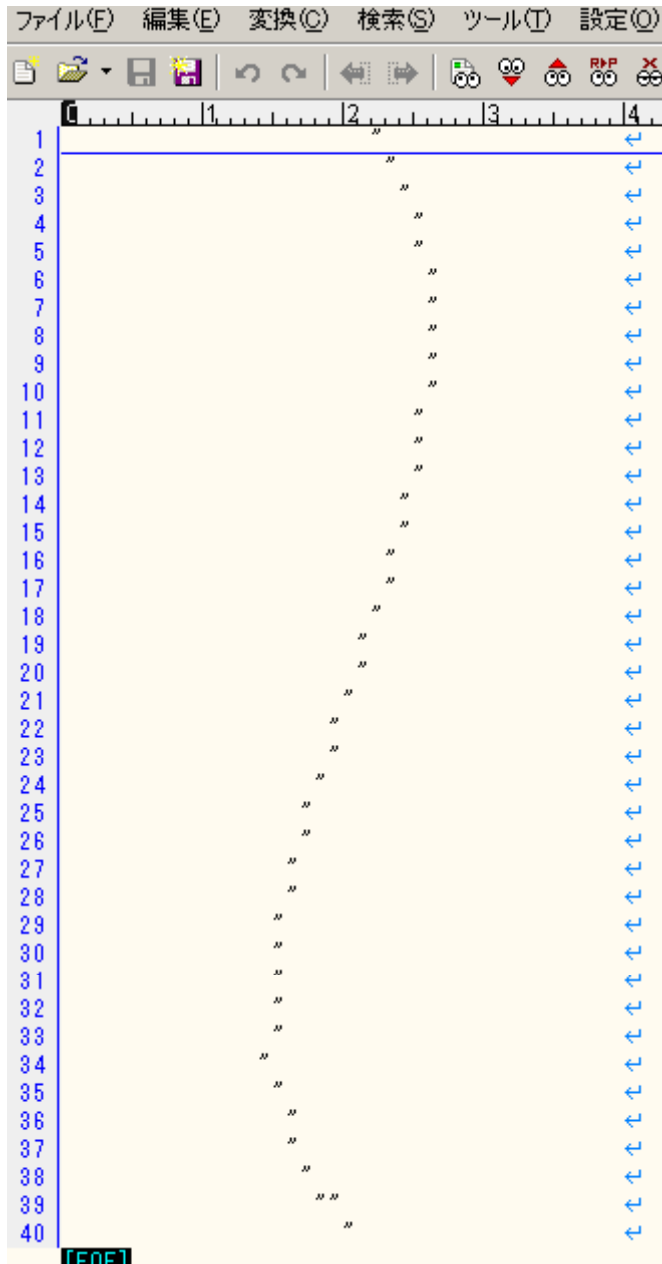
または 2 バイトのシフト JIS 文字を認識できます。すなわち、偽として、全角の空白を利用できます。偽ならば、白い点になります。偽ではない文字は、真であると認識します。たとえば、四角( )は、真であると認識されて、黒い点になります。

#### 必須ではない true.txt

黒い点に対応させたい文字を true.txt の各行に記入してください。真として、#を記入する必要はありません。Kopic は、1 バイトまたは 2 バイトのシフト JIS 文字を認識できます。すなわち、真として、全角の四角( )を利用できます。真ならば、黒い点になります。false.txt と異なり、true.txt は、必須ではありません。

#### 一時ファイル

true.txt を開くことができた場合、Kopic は、真を示す文字を引用符に置換して、一時ファイルを作成します。tru 番号-分-秒.txt の形式で一時ファイルを命名します。次回、**Kopic は、まず一時ファイルを削除して、次に命令を実行します。** サクラエディタで一時ファイルを表示した例



### テキストファイルから XBM を作成

kopic.txt から kopic.xbm を作成するには、コマンドプロンプトに下記のコマンドを入力して、Enter を押してください。

```
java Kopic kopic.txt
```

### テキストファイルから XBM を作成して高速選択

kopic.txt から kopic.xbm を作成して、即座に新しい kopic.xbm で高速選択を行うには、-n を指定してください。一時ファイルが作成されます。高速選択は、コマンドプロンプトに java Kopic のみ入力して Enter を押した場合と同じ結果になります。-n が指定され

たかどうかによる相違を図示します。

```
C:¥KopicJava>java Kopic kopic.txt -n  
pic57-1-44.txt
```

```
C:¥KopicJava>java Kopic kopic.txt
```

```
C:¥KopicJava>java Kopic  
pic58-1-51.txt
```

### ファイル名の拡張子

Kopic は、ファイル名の拡張子として、.csv と .kun と .xbm を認識できます。それ以外は、テキストファイルであることを想定します。kopic.csv を指定した場合、kopic.xbm は、無視されます。バックアップの章 ([r.csv](#)) を参照してください。

| ファイル名の拡張子          | 実行       | 出力             |
|--------------------|----------|----------------|
| .csv               | ロト7の高速選択 | Pic 番号-分-秒.txt |
| .kun               | 画像を復元    | kopic.txt      |
| .xbm               | ロト7の高速選択 | Pic 番号-分-秒.txt |
| その他(拡張子が無いファイルを含む) | XBM を作成  | kopic.xbm      |

### XBM からテキストファイルを作成

kopic.xbm は、テキストファイルですが、十六進数の配列であり、わかりにくい。画像を概観するために、kopic.xbm から kopic.txt を作成するには、コマンドプロンプトに下記のコマンドを入力して、Enter を押してください。

```
java Kopic t
```

## バージョン表示

ファイル名の拡張子として .xbm を有しているファイルが存在しないフォルダで、Kopic をテストするために、kopic.xbm(切れ目がない画像)を作成するには、コマンドプロンプトに下記のコマンドを入力して、Enter を押してください。

```
java Kopic y
```

### 一時ファイルを削除

Kopic は、まず一時ファイルを削除します。一時ファイルとしては、error.txt および pic 番号-分-秒.txt があります。たとえば、pic12-34-56.txt と同一の内容が Kopic のログ (KO.LOG) に含まれていますが、pic12-34-56.txt を削除してはならない場合、-r-を指定してください。



## 短いコメント

-V または -VC が指定された場合、Kopic は、バージョン番号を出力します。短いコメントを付加するには、-VC を指定してください。

バージョン番号および短いコメントを出力した例

```
C:\¥KopicJava>java Kopic -VC
Kopic Java Version 1.13

java Kopic          # Read kopic.xbm
java Kopic -c       # Produce kopic.csv from kopic.xbm
java Kopic -c -38-  # Use From, not 38.
java Kopic -cd2     # Produce kopic.csv from kopic.xbm and decre
java Kopic -e-      # Do not edit kopic.csv
java Kopic -f       # Produce full-swing kopic.txt from kopic.cs
java Kopic -g       # Use rannu.txt to weight balls.
java Kopic -k       # Produce kopic.kun from kopic.xbm
java Kopic -m       # Read many_one.txt
java Kopic -o30     # Output 30 lines.
java Kopic -r-      # Do not remove temporary files.
java Kopic -t       # Produce kopic.txt from kopic.xbm
java Kopic -y       # recovery
java Kopic -V       # version
java Kopic -VC      # version with comments
java Kopic -w-      # Do not weight balls.
java Kopic file_name.csv # Read file_name.csv regardless of
java Kopic file_name.kun # Produce kopic.txt from file_name
java Kopic file_name.txt # Produce kopic.xbm from file_name
java Kopic file_name.xbm # Read file_name.xbm
java Kopic file_name.txt -n # Produce and read new kopic.xbm
```

## 設定ファイル

-e- が指定された場合、Kopic は、kopic.xbm を読み込みません。kopic.csv の最後にある 4 行は、kopic.xbm に基づいていません。すなわち、kopic.csv は、ロト7高速選択ソフトウェア (Kopic) に必須の設定ファイルです。kopic.csv を開くことができない場合、Kopic は、自動的に kopic.csv を作成します。-c を指定した場合と同じことになります。

**kopic.csv の内容を確認してください。**たとえば、Setting を 0 に設定すれば、年月日時分秒により、実行するたびに異なる結果になります。

## アイドリング回数

From は、乱数発生器のアイドリング回数です。0 以上の整数を設定できます。アイドリング状態であるときに発生した乱数は、Kopic により無視されます。

### 38 ではなく From を書き出したい

-c のみ指定した場合、From ではなく 38 が書き出されます。kopic.csv の内容を確認してください。38 が書き出された場合、Kopic で高速選択を行うとき、自動的に From の行が生成されます。38 ではなく From が書き出されるように、-38-を併用できます。

## java Kopic -c -38-

### 乱数表

-w-または-mまたは-g が指定されたにもかかわらず、乱数表として rannu.txt を開くことができない場合、Kopic は、自動的に 768KB の rannu.txt を作成して続行します。C 版の rannu コマンドが Kopic に添付されています。

### メルセンヌツイスタ

乱数発生器として、MTRandom が使用されています。MTRandom は、ライブラリとして Java で実装されたメルセンヌツイスタ (MT) です。

MTRandom Copyright © 2005, David Beaumont

### ログファイル

KO.LOG など、ログファイルの名前を Log に設定します。.LOG や.txt など、ファイル名の拡張子が必要です。ログファイルを開くことができない場合、Kopic は、新しいログファイルを作成します。テキストエディタでログファイルを開いて、日時および追加行数をヒントにして該当部分を発見できます。

### 一時ファイル

Kopic は、一時ファイルを出力しますが、一時ファイルと同一の内容をログに追加しません。Kopic は、pic 番号-分-秒.txt の形式で一時ファイルを命名します。次回、Kopic は、まず一時ファイルを削除して、次に高速選択を実行します。

### 乱数発生器を駆動する回数

Repeat は、乱数発生器を駆動する回数であり、アイドル回数を含みます。すなわち、Repeat は、From より大きい整数です。

### 何とおりの組合せを出力するか

たとえば、5 とおりの組合せを出力するには、Repeat を下記の式で設定します。組合せの数は、5 であり、5 行が出力されます。

$$\text{Repeat} = \text{From} + 5$$

### -o 組合せの数

From および Repeat の両方が CSV ファイルに記入されている場合、しかも、CSV ファイルが指定されている場合、Kopic は、CSV にしたがって、高速選択を行います。CSV

ファイルに不備があるにもかかわらず、利用者が CSV ファイルを指定した場合、しかも、-o 組合せの数 (-o30 など) を指定した場合、Kopic は、自動的に上記の式に適合するように、組合せの数にしたがって、Repeat または From を調節します。

## 乱数発生器の初期状態

乱数発生器は、擬似乱数を発生しますが、たとえば、123456 などの整数で初期状態を変更できます。1 から 999999 までの整数を想定しています。

### 再現

真の乱数と異なり、擬似乱数は、コンピュータで再現できる数列です。Setting を変更することで、選択結果を変更できますが、変更しないかぎり高速選択を何回、実行しても、選択結果は、同じになります。

### 年月日時分秒

Setting が 0 である場合、Kopic も rannu コマンドも、自動的に年月日時分秒に基づく整数で乱数発生器の初期状態を変更します。実行するたびに時刻が異なるため、選択結果が変化します。

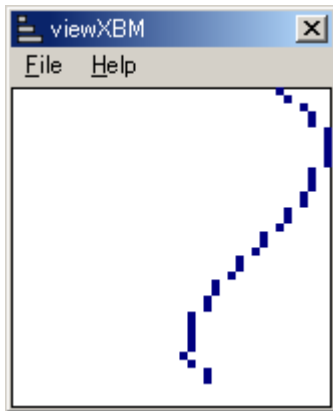
## コメント

CSV の第 2 欄および第 3 欄の間には、コンマがあります。シャープ、ナンバーサイン、ハッシュ ( # ) をコンマの直後に挿入することにより、利用者は、CSV の第 3 欄に **コメントを記入できます**。ただし、Kopic は、**コメントを削除して、新しい kopic.csv を作成します**。

# 32,0,# ミニロト状態

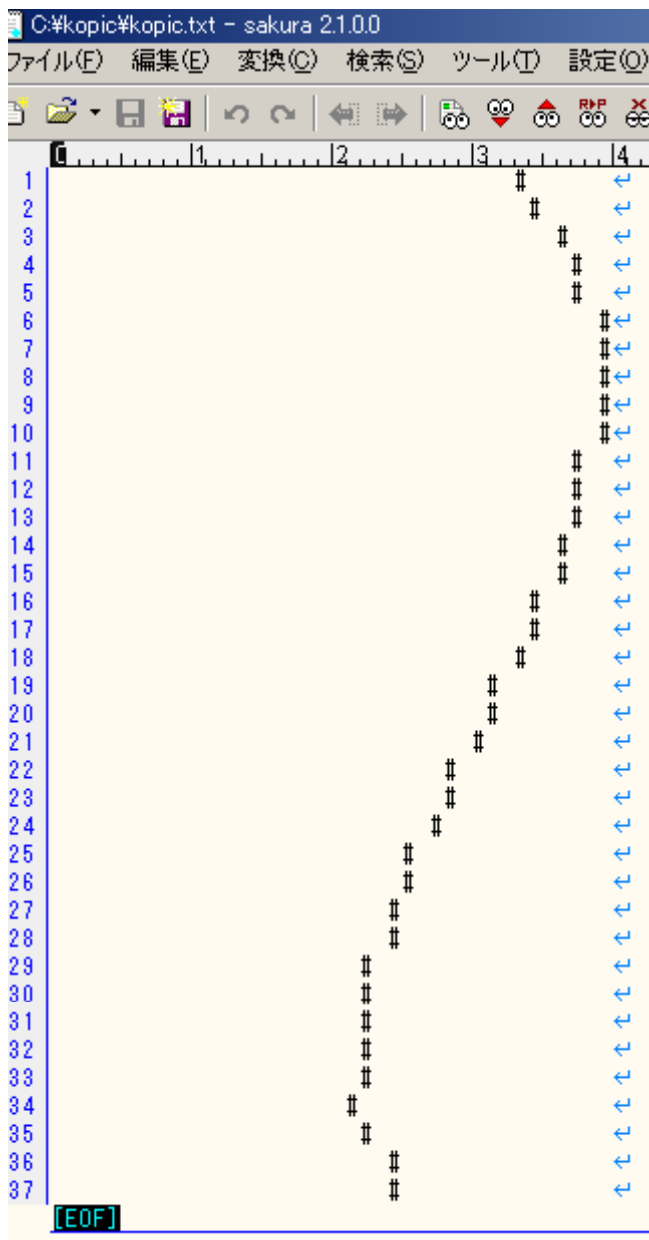
## フルスイング

-f が指定された場合、重みの最大値が 39 になるように、Kopic は、CSV ファイルを修正して、新しい kopic.csv を作成します。さらに、kopic.txt を作成します。フルスイングの kopic.txt から作成された kopic.xbm を viewXBM で描画した例



### CSV から XBM を作成

-f を指定することでフルスイングの kopic.txt を作成できます。



### XBM を作成して即座に実行した例

XBM を復旧して、kopic.csv を作成して、フルスイングの kopic.txt を作成して、そのテキストファイルから XBM を作成して、高速選択を即座に実行した例

```
C:¥KopicJava>java Kopic -y
C:¥KopicJava>java Kopic -c
C:¥KopicJava>java Kopic -f
pic55-23-14.txt
C:¥KopicJava>java Kopic kopic.txt -n -r-
pic56-23-23.txt
C:¥KopicJava>FC pic55-23-14.txt pic56-23-23.txt
ファイル pic55-23-14.txt と PIC56-23-23.TXT を比較しています
FC: 相違点は検出されませんでした
```

## バックアップ

kopic.csv は、Kopic によって変更されます。kopic.csv のバックアップを行ってください。kopic.csv ではない名前を付けて保存してください。たとえば、kopic.csv を r.csv にコピーしてください。Kopic は、ファイル名の拡張子として.csv を認識できません。kopic.csv ではなく r.csv を指定して高速選択を実行するには、コマンドプロンプトに下記のコマンドを入力して、Enter を押してください。

## java Kopic r.csv

### kopic.xbm を無視

CSV ファイルが指定された場合、Kopic は、XBM を読み込まないで、高速選択を実行します。

### kopic.csv が変更される例

10 番台が出現しないように設定するために、利用者は、テキストエディタで kopic.csv を開いて、10 から 19 までの行を削除できます。コマンドプロンプトに下記のコマンドを入力して、Enter を押してください。Kopic は、自動的に kopic.csv の 10 から 19 まで 0 を記入します。

## java Kopic kopic.csv

### 空白が削除される

見やすいように、利用者は、CSV に空白を挿入できます。ただし、Kopic は、自動的に kopic.csv から空白を削除します。

### kopic.csv の変更を禁止

Kopic は、正しく機能しやすいように、r.csv および kopic.csv などの CSV ファイルを検査して編集しますが、kopic.csv を編集してはならない場合、CSV ファイルを指定しな

いで、-e-を指定してください。kopic.xbm を読み込まないで、Kopic は、完璧な CSV ファイルとして kopic.csv を使用します。CSV ファイルおよび-e-の両方が指定された場合、-e-は、効果がありません。

## java Kopic -e-

### 乱数表の数字からなるロト7

-w-が指定された場合、Kopic は、kopic.csv による重み付与を行わないで、rannu.txt の数字からなるロト7を出力します。すなわち、乱数表の数字を順に読み込んで、重複が無いように7個の数字を組み合わせ、昇順に並べ替えて各行を出力します。

#### 乱数表ソフトウェア

rannu コマンドは、乱数表を作成します。乱数は、二桁の十進数です。**最小値、最大値を設定できます**(たとえば、1 から 37 まで)。乱数表のサイズを指定できます。標準出力でリダイレクトを利用できます。乱数の種を設定できます。rannu コマンドは、メルセンヌツイスタ(MT)を利用する 32 ビットアプリケーションです。

#### rannu コマンドをインストール

1. ロトサマリーというウェブサイトから rannu-C.zip をダウンロードしてください。
2. ZIP を展開してください。
3. 利用者は、フォルダの改名および移動が可能です。

```
C:\KopicJava>rannu -VC
rannu C-1.06

rannu          # Produce rannu.txt
rannu -6       # Loto-6
rannu -7       # Loto-7
rannu -c       # CSV
rannu -f400    # idling
rannu -g       # interval of 0 to 63
rannu -i1-37   # interval of 1 to 37
rannu -k32     # 32 kilobytes
rannu -m       # Miniloto
rannu -r       # standard output
rannu -r > file_name # Redirect.
rannu -s12345  # seed
rannu -s0      # Time is used as seed.
rannu -V       # version
rannu -VC      # version with comments
```

### 0 から 63 までの乱数表を利用して重みを付与

-g が指定された場合、rannu コマンドは、MT から 32 ビットの乱数を受領して、6 ビット

ずつ利用して、0から63までの乱数表を作成します。-gが指定された場合、Kopicは、内蔵のMTで乱数を発生するのではなく、0から63までの乱数表を読み込んで、32ビットずつ利用して、0から4294967295までの乱数を作成します。kopic.csvにしたがって、Kopicは、ボールの重みを考慮しながら、ロト7の組合せを出力します。



| パラメータ | 乱数の発生源 | 乱数表       | 重みを付与     | 多対一の対応       |
|-------|--------|-----------|-----------|--------------|
| なし    | 内蔵の MT | 不要        | kopic.csv |              |
| -w-   | 不要     | rannu.txt | 重みを付与しない  |              |
| -m    | 不要     | rannu.txt |           | many_one.txt |
| -g    | 不要     | rannu.txt | kopic.csv |              |

## 多対一の対応

-m を指定する前に、乱数から数に変換する表を用意してください。many\_one.txt に表を記載することで、乱数表を変換できます。たとえば、一桁の数を十台の数に対応させて、十台の数は、元通りの十台の数に対応させた場合、一対一の対応ではなく二対一の対応になります。多対一の対応を意味する変換表の罫線として、空白、縦線、空白の 3 文字を利用してください。

ファイル(F) 編集(E) 変換(C) 検索(S)

📄 🗑️ 📁 📄 ↺ ↻ ⏪ ⏩

0 1 2

|    |    |    |   |
|----|----|----|---|
| 1  | 01 | 11 | ↵ |
| 2  | 02 | 12 | ↵ |
| 3  | 03 | 13 | ↵ |
| 4  | 04 | 14 | ↵ |
| 5  | 05 | 15 | ↵ |
| 6  | 06 | 16 | ↵ |
| 7  | 07 | 17 | ↵ |
| 8  | 08 | 18 | ↵ |
| 9  | 09 | 19 | ↵ |
| 10 | 10 | 10 | ↵ |
| 11 | 11 | 11 | ↵ |
| 12 | 12 | 12 | ↵ |
| 13 | 13 | 13 | ↵ |
| 14 | 14 | 14 | ↵ |
| 15 | 15 | 15 | ↵ |
| 16 | 16 | 16 | ↵ |
| 17 | 17 | 17 | ↵ |
| 18 | 18 | 18 | ↵ |
| 19 | 19 | 19 | ↵ |
| 20 | 20 | 20 | ↵ |
| 21 | 21 | 21 | ↵ |
| 22 | 22 | 22 | ↵ |
| 23 | 23 | 23 | ↵ |
| 24 | 24 | 24 | ↵ |
| 25 | 25 | 25 | ↵ |
| 26 | 26 | 26 | ↵ |
| 27 | 27 | 27 | ↵ |
| 28 | 28 | 28 | ↵ |
| 29 | 29 | 29 | ↵ |
| 30 | 30 | 20 | ↵ |
| 31 | 31 | 21 | ↵ |
| 32 | 32 | 22 | ↵ |
| 33 | 33 | 23 | ↵ |
| 34 | 34 | 24 | ↵ |
| 35 | 35 | 25 | ↵ |
| 36 | 36 | 26 | ↵ |
| 37 | 37 | 27 | ↵ |

[EOF]