

固定長フォーマットデータのための簡易集計ユティリティ (ver.2.1)  
2011-02-04 修正版

大津起夫\*

\* 大学入試センター 研究開発部 試験評価解析研究部門  
Department of evaluation and analysis  
Research and Development Division,  
The National Center for University Entrance Examinations

## 固定長フォーマットデータのための簡易集計ユティリティ

大津起夫<sup>1</sup>

### 要約

大学入試センター業務および研究開発部での分析作業においては、大量の固定長データの集計処理を行う必要が頻繁に生じる。現状では、これらの分析には統計ソフトウェアが用いられることが多い。このうち、大量データの処理が必要になるのは、分析目的に応じた項目（フィールド）の切り出しとその集計作業である。この部分の処理をPC上で簡便に行うためのユティリティ群を開発した。これらを利用することにより、従来統計ソフトウェアを用いていた大量データ処理を、ライセンスに気遣うことなく全てのPC上で利用できる。また、項目反応モデルなど特定の分析目的用のソフトウェアや表計算ソフトウェアへの入力用データを簡単なスクリプトによって作成できる。

### キーワード

固定長データ、ファイル処理、C++、STL

*Simple Utilities for Counting Fixed-Formatted Records on PC*

Tatsuo OTSU<sup>1</sup>

### Abstract

Nation-wide test data analysis frequently requires handling of large number of records in fixed formatted files. Some simple utility programs on PC were developed. These programs are able to pick-up selected fields, count up keys, merge files, and select specified records. Their simplicity enables fast large-scale data processing.

---

<sup>1</sup>Research and Development Division, The National Center for University Entrance Examinations

# 1 はじめに

センター入試業務および研究開発部における分析作業においては、アスキーコードによって記述された大量の固定長フォーマットデータの処理の必要が頻繁に生ずる。これらの作業の多くは、共通のパターンを持っているが、大量データに対応したアプリケーションソフトウェアが少ないため、単純な前処理作業のために高機能な統計ソフトウェアを用いるか、あるいは専用のプログラムを作成せざるを得ない場合が多い。

これらのファイル処理に共通な作業を大量データについて実行できるユティリティ群をPC上で利用可能とすれば、ライセンスの問題に気遣うことなく、センター内の任意のPC上でデータ処理を可能とすることができる。多くの場合データの分析作業は

1. 関連するフィールドの切り出し
2. 基礎統計量の計算
3. これらに基づくモデル当てはめ

のフェーズを持つが、大量データの取り扱いが必要であるのは多くの場合1と2であり、これらの処理においては高機能な統計ソフトウェアの能力はほとんど必要とされない。

ここではこれらのデータ前処理を行うユティリティプログラム群を開発した。これらを利用することにより、簡単なスクリプトによって表計算ソフトウェアや特定のモデル推定用のソフトウェアへの入力用データを準備することができる。作成したプログラム群は、アスキーコードによって記述された固定長フォーマットデータを対象として、次の処理を行う。

1. フィールド（カラム）の切り出し
2. 同一のキー値を持つレコード件数の算出
3. キー値による2つのファイルのマージ
4. 指定カラムの値によるレコード選択

各プログラムはC言語および一部C++言語によって記述されている。コンパイルはWindowsXP上のインテル社製C++コンパイラ(ver 10.1)でおこなった。処理系は比較的高価だが、大量データの処理についての性能が良いため採用した。また、第1版において入出力を1バイト単位の繰り返しループで行っていた部分を、ライブラリ関数を利用するように変更することにより、一部の関数(fixedEdit, fixedAndなど)で処理速度が向上した。また頻度集計のためにSTLのmapコンテナを用いている。

コンパイルされた各プログラムはMicrosoft社製のWindowsXPとVistaのコマンドプロンプト(CMD)ウインドウ上で動作することを確認してある。ファイル名などの指定は、コマンドの引数として与える。またカラムの認識はバイト単位であり、ワイドキャラクタを1文字単位で処理する対応は行っていない。また、各ファイルは固定長であることを前提としているが、レコードの終わりの認識は行末コード{Windowsの場合はCR(0x0D)とLF(0x0A)}によって行っており、行の途中にこれらの値がデータとして含まれることを許さない。

プログラム名と機能の一覧を表1に示す。プログラムは、ほぼANSI準拠の機能によって記述しているが、一部Windows固有のマクロを用いている部分もあり、他OSへの移植時に注意すべき点がある。

表 1: プログラム名と機能一覧

プログラム名	機能
fixedEdit	* フィールド (カラム) の切り出し
fixedCount	* ソート済みファイルにおけるキー値毎のレコード件数の算出 (指定ファイルへ出力)
fixedCnt	* ソート済みファイルにおけるキー値毎のレコード件数の算出 (標準出力へ出力)
fixedMerge	* キー値により 2 つのソート済みファイルをマージ (マッチングされたレコードのみ出力)
fixedMergeF	* 同上 (マッチングされないレコードも出力)
fixedAnd	* カラムの値によるレコードの選択 (AND 条件の指定)
fixedOr	* 同上 (OR 条件の指定)
fixedAndC	* カラムの値によるレコードの選択 (1 条件のみ)
fixedKeyDel	指定ファイル中にあるキー値を含むレコードの削除
fixedKeySel	指定ファイル中にあるキー値を含むレコードの選択
fixedFreqs	カラム値の頻度と相対頻度を求める
fixedLfreqs	カラム値の層別頻度を求める
fixedDupl	重複レコードの検出
fixedIns	指定カラムへの文字列の挿入
asctime	現在日時を秒単位で表示
prproof	* 行範囲を指定しての表示 (表示幅 70 カラム)
prproof100	* 行範囲を指定しての表示 (表示幅 100 カラム)
fixedFldfreqs2	複数のフィールドの頻度集計
fixedFldlfreqs2	複数のフィールド対の頻度集計

一部のプログラム (\* 印) では入力レコード長が最大 8196 バイトに制限される。

次節では、各プログラムの機能を順次解説する。各プログラムの説明の「利用法」の項において、かぎ括弧 [ ] はコマンドライン引数が省略可能であることを示す。

## 2 プログラム機能の解説

### 2.1 fixedEdit

#### 利用法

```
fixedEdit 入力カラム情報ファイル名 出力カラム情報ファイル名
```

#### 機能

入力ファイルからとりだすカラムを入力カラム情報ファイルに指定し、各レコードの指定された部分を標準出力に出力する。固定長データは標準入力から読み込み、出力ファイルのカラムフォーマットは出力カラム情報ファイルに記録される。入力カラム情報ファイルおよび出力カラム情報ファイルをコマンド行の引数で指定する。

#### コマンドライン引数

##### (1) 入力カラム情報ファイル

カラム編集の指定を行う。複写開始カラム（整数）と複写するカラム長（正の整数）および変数名（文字列）の3つを空白で区切って各行に指定する（全角のスペースは不可）。変数名は空白文字を含まなければ8バイトコードを含んでもよいが、ワイドキャラクタへの対応を特別には行っていない。先頭第1カラムがアスタリスク（\*）の行は、コメントとみなされる。また空白文字のみの行（スペース、タブ、復改のみの行）は無視される。複写開始カラムは1からレコード長（行末コードを含まない）の範囲でなければならない。ただし、複写開始カラムを0とすると、カラム長に指定された長さの空白を出力する。また複写開始カラムを-1とするとカンマ（,）が出力される。また-2を指定するとTABコードが出力される。出力カラムは、複写されたカラム長から自動的に計算される。複写開始カラムは、昇順に指定する必要はない。

##### (2) 出力カラム情報ファイル

出力ファイルのカラムについての情報が記録される。入力カラム情報ファイルの1カラムがアスタリスク（\*）の行（コメント行）は先頭にアスタリスクをさらに1文字追加して出力カラム情報ファイルに出力される。

#### 標準入出力

##### (1) 標準入力

固定長のテキストファイル。行末には改行コードが必要。Microsoft Windows の場合は CR+LF、つまり16進数で0Dと0Aの2バイトの列が改行コードとなる。Linux/Unix や、Mac では改行コードは異なるが、そのOS用のGCCでソースコードがコンパイルされていれば、改行コードを適切に扱える。）

(2) 標準出力 編集されたデータが出力される。出力行数は入力行数と同じ。

(3) 標準エラー出力

入力レコード件数、および異常時のメッセージが出力される。

## 利用例

```
fixedEdit test1.clm test1out.clm < test1.dat > test1out.dat 2> test1.log
          (1)         (2)         (3)         (4)         (5)
```

ここで(3)の test1.dat の前の<は標準入力のリダイレクション(通常はコンソールへの出力をファイルへ割り当てること)を、(4)の test1out.dat 前の>は標準出力のリダイレクションを表す。また、(5)の test1.log の前の、2> は標準エラー出力のリダイレクションを示す。

(1) test1.clm 入力カラム情報ファイル

```
-----サンプルの開始-----
* 入力カラム カラム長 変数名
9 2 第9カラム
0 2 空白1
7 2 第7カラム
-1 1 コンマ
1 5 第1カラム

-----サンプルの末尾-----
```

(2) test1out.clm 出力カラム情報ファイル

```
-----サンプルの開始-----
*InputColumnInfo test1.clm
*OutputColumnInfo test1out.clm
*      Out      Len  Varname
** 入力カラム カラム長 変数名
      1          2  第9カラム
      3          2  空白1
      5          2  第7カラム
      7          1  コンマ
      8          5  第1カラム

-----サンプルの末尾-----
```

(3) test1.dat 入力データ(標準入力)

```
-----サンプルの開始-----
0407934712
0760443210
7823440052
8115555394
9544139358
9959494198
6667174394
1522332756
1729750479
....
....

-----サンプルの末尾-----
```

(4) test1out.dat 出力ファイル(標準出力)

```

-----サンプルの開始-----
12 47,04079
10 32,07604
52 00,78234
94 53,81155
58 93,95441
98 41,99594
94 43,66671
....
-----サンプルの末尾-----

```

(5) test1.log 実行時メッセージ (標準エラー出力)

```

-----サンプルの開始-----
*fixedEdit:InputRecords 100
-----サンプルの末尾-----

```

## 2.2 fixedCount

### 利用法

fixedCount 出力ファイル名 [キー開始カラム [キーの長さ]]

### 機能

ソート済みのファイル中の、同一のキーを持つレコード件数を求める。

### コマンドライン引数

- (1) 出力ファイル名  
出力を行うファイルを指定する。指定されたカラム位置に対応するキー値とその件数 (カラム幅 12) が各行に出力される。
- (2) キー開始カラム  
キーの開始カラム (バイト) を正の整数で指定する。
- (3) キーの長さ  
キーの長さ (バイト) を正の整数で指定する。  
「キーの開始カラム」と「キーの長さ」をともに省略すると、レコード全体をキーとみなす。「キーの長さ」を省略すると、キー開始位置から行末までをキーとみなして集計を行う。

### 標準入出力

- (1) 標準入力  
指定されるキー部分についてソート済みの固定長データを含むものとする。
- (2) 標準出力  
標準出力には処理レポート (キー指定) が表示される。
- (3) 標準エラー出力  
エラーメッセージおよび入出力件数が表示される。

## 利用例

```
sort /+9 test1.dat | fixedCount test1.cnt 9 1
(1)      (2)      (3)              (4)  (5)(6)
```

- (1) sort /+9  
Windows のソートユーティリティ (Linux や MinGW の sort コマンドでは引数の指定法が異なる) を用いて 9 カラム以降をキーとして整列する。結果は標準出力に出力される。
- (2) test1.dat  
入力ファイル名を表す。
- (3) |  
パイプ処理を表す。ソートされたデータが fixedCount の標準入力となる。
- (4) test1.cnt

出力ファイル名を示す。次のものが得られる。

```
キー      件数
-----サンプルの開始-----
0          8
1          6
2          2
3         12
4         11
5         11
6         17
7          7
8         11
9         15
-----サンプルの末尾-----
```

- (5) キー開始カラム
- (6) キーの長さ
- (7) 標準出力  
出力ファイル名とキー指定の確認が表示される。

```
-----サンプルの開始-----
*Outputfile test1.cnt
*KeyStartPos 9
*KeyLength 1
-----サンプルの末尾-----
```

- (8) 標準エラー出力  
エラー報告と入出力件数が表示される。

```
-----サンプルの開始-----
*InputRecords 100
*OutputRecords 10
-----サンプルの末尾-----
```

## 2.3 fixedCnt

### 利用法

```
fixedCnt キー開始カラム [キーの長さ]
```

### 機能

fixedCount と同様に、ソート済みのファイル中の、同一のキーを持つレコード件数を求める。ただし、集計件数は標準出力に出力される。

### コマンドライン引数

- (1) キー開始カラム  
キーの開始カラム (バイト) を正の整数で指定する。
- (2) キーの長さ  
キーの長さ (バイト) を正の整数で指定する。  
「キーの開始カラム」と「キーの長さ」をともに省略すると、レコード全体をキーとみなす。「キーの長さ」を省略すると、キー開始位置から行末までをキーとみなして集計を行う。

### 標準入出力

- (1) 標準入力  
指定されるキー部分についてソート済みの固定長データを含むものとする。
- (2) 標準出力  
指定されたカラム位置に対応するキー値とその件数 (カラム幅 12) が各行に出力される。
- (3) 標準エラー出力  
処理レポート (キー指定)、入出力件数およびエラーメッセージが表示される。

### 利用例

```
sort /+9 test1.dat | fixedCnt 9 1 > test1std.cnt
```

処理内容は fixedCount と同様。

## 2.4 fixedMerge/fixedMergeF

### 利用法

```
fixedMerge 第1入力ファイル名 第1カラム情報ファイル名 第2入力ファイル名  
( 行の続き ) 第2カラム情報ファイル名 出力カラム情報ファイル名  
( 行の続き ) [ キー開始カラム1 キー開始カラム2 キーの長さ ]
```

```
fixedMergeF 第1入力ファイル名 第1カラム情報ファイル名 第2入力ファイル名  
( 行の続き ) 第2カラム情報ファイル名 出力カラム情報ファイル名  
( 行の続き ) [ キー開始カラム1 キー開始カラム2 キーの長さ ]
```

## 機能

ソート済みの2つのファイルを入力し、キーの値によってマージ処理を行う。それぞれのファイルにおいて、キーの値は一意的でなければならない。fixedMerge はマッチングされたレコードのみを出力し、fixedMergeF はマッチングされないレコードをも、欠落部分にピリオドを埋めて出力する。キーの値がファイル中で一意でない場合、あるいはキー値についてソートされていない場合には、動作は保障されない。

## コマンドライン引数

- (1) 第1入力ファイル  
一意のキーを持つ固定長レコードからなり、キー値について昇順にソート済みでなければならない。
- (2) 第1カラム情報ファイル  
各行に開始カラム、カラム長、および変数名の3つを含み、第1入力ファイルのフォーマットを記述する。第1カラムがアスタリスク(\*)の行はコメントとみなされる。このファイルには、第1入力ファイルのすべてのフィールドが記述されていなければならない。このファイルの内容は、マージ操作には用いられないが、出力カラム情報ファイルを作成するために用いられる。
- (3) 第2入力ファイル  
第1入力ファイルと同様
- (4) 第2カラム情報ファイル  
各行に開始カラム、カラム長、および変数名の3つを含み、第2入力ファイルのフォーマットを記述する。第1カラムがアスタリスク(\*)の行はコメントとみなされる。このファイルには、第2入力ファイルのすべてのフィールドが記述されていなければならない。このファイルの内容は、マージ操作には用いられないが、出力カラム情報ファイルを作成するために用いられる。
- (5) 出力カラム情報ファイル  
2つの入力ファイルをキーの値によってマージした出力ファイルのカラム情報が記録される。前半のカラムが第1入力ファイル、後半のカラムが第2入力ファイルの内容となる。第1入力ファイルの内容と第2入力ファイルの内容とが併せて出力される。第1カラム情報ファイルおよび第2カラム情報ファイル中のコメント行は先頭にアスタリスクを1個追加して出力される。
- (6) キー開始カラム1  
第1入力ファイルにおけるキーの開始位置。正の整数で指定する。
- (7) キー開始カラム2  
第2入力ファイルにおけるキーの開始位置。正の整数で指定する。
- (8) キーの長さ  
キーのカラム長を指定する。正の整数が指定された場合には、キーの値を確認してマージを行う。(6),(7),(8)が省略された場合には入力の順に無条件でマッチングを行う。

## 標準入出力

- (1) 標準入力  
標準入力は使用しない。
- (2) 標準出力  
マージされたレコードが出力される。キーは同一の値が1レコード中に重複して現れる。fixedMerge ではマッチングされるべきレコードがない場合には、レコードは出力されない。fixedMergeF では、欠落部分がピリオドで埋めて出力される。
- (3) 標準エラー出力  
エラー情報を表示する。あわせてそれぞれの入力レコード件数とマッチされなかったレコード件数およびマッチングに成功したレコード件数も記録される。

#### 利用例 1

```
fixedMerge testm1.dat testm1.clm testm2.dat testm2.clm
           (1)      (2)      (3)      (4)
( 行の続き ) testmout.clm 1 3 2 >testmout.dat 2> testm.log
           (5)                        (6)      (7)
```

##### (1) testm1.dat 第1入力ファイル

```
-----サンプルの開始-----
01aA
02bB
03cC
04dD
05eE
06fF
07gG
08hH
09iI
10jJ
-----サンプルの末尾-----
```

##### (2) testm1.clm 第1カラム情報ファイル

```
-----サンプルの開始-----
* testm1.dat
1 2 testm1Key
3 2 testm1Body
-----サンプルの末尾-----
```

##### (3) testm2.dat 第2入力ファイル

```
-----サンプルの開始-----
zz00AA
yy02BB
xx04DD
ww05EE
vv06FF
uu07GG
tt08HH
ss09II
rr11JJ
-----サンプルの末尾-----
```

(4) testm2.clm 第2カラム情報ファイル

```
-----サンプルの開始-----  
* testm2.dat  
1 2 testm2dummy  
3 2 testm2Key  
5 2 testm2Body
```

```
-----サンプルの末尾-----
```

(5) testmout.clm 出力カラム情報ファイル

```
-----サンプルの開始-----  
*Infile1      testm1.dat  
*ColInfoFile1 testm1.clm  
*Infile2      testm2.dat  
*ColInfoFile2 testm2.clm  
*Outfile      stdout  
*OutColInfoFile testmout.clm  
*      Out      Len  Varname  
** testm1.dat  
      1          2  testm1Key  
      3          2  testm1Body  
*      Out      Len  Varname  
** testm2.dat  
      5          2  testm2dummy  
      7          2  testm2Key  
      9          2  testm2Body  
*KeyStart1    1  
*KeyStart2    7  
*KeyLength    2  
*RecordLength1 4  
*RecordLength2 6  
*OutRecordLength 10  
-----サンプルの末尾-----
```

(6) testmout.dat 標準出力

```
-----サンプルの開始-----  
02bByy02BB  
04dDxx04DD  
05eEww05EE  
06fFvv06FF  
07gGuu07GG  
08hHtt08HH  
09iIss09II  
-----サンプルの末尾-----
```

(7) testm.log 標準エラー出力

```
-----サンプルの開始-----  
*InputRecords1 10  
*Unmatched1    3  
*InputRecords2 9  
*Unmatched2    2  
*Matched       7  
-----サンプルの末尾-----
```

利用例2 マッチングしないレコードも出力する例

```
fixedMergeF testm1.dat testm1.clm testm2.dat testm2.clm
```

(行の続き) testmoutF.clm 1 3 2 >testmFout.dat 2> testmF.log  
(5') (6') (7')

(5') testmoutF.clm 出力カラム情報

testmout.clm と同一になる。

(6') testmoutF.dat 標準出力

```
-----サンプルの開始-----  
...zz00AA  
01aA.....  
02bByy02BB  
03cC.....  
04dDxx04DD  
05eEww05EE  
06fFvv06FF  
07gGuu07GG  
08hHtt08HH  
09iIss09II  
10jJ.....  
...rr11JJ  
-----サンプルの末尾-----
```

(7') testmF.log 標準エラー出力

```
-----サンプルの開始-----  
*InputRecords1 10  
*Unmatched1 3  
*InputRecords2 9  
*Unmatched2 2  
*Matched 7  
-----サンプルの末尾-----
```

## 2.5 fixedAnd/fixedOr

### 利用法

fixedAnd 選択条件ファイル名

fixedOr 選択条件ファイル名

### 機能

選択条件ファイルの各行よりレコード選択の条件を読み込み、標準入力から入力したレコードのうち、それらの条件をすべて満たしたレコード (fixedAnd の場合) あるいはいずれかの条件を満たしたレコード (fixedOr の場合) を選択し、標準出力に出力する。

### コマンドライン引数

#### (1) 選択条件ファイル

各行に、指定開始カラム (整数) 条件演算子 値 の3つを空白で区切って指定する。条件演算子は、値とフィールドの比較に ==, !=, >=, <= (または =<), >, < の6種類が指定可能。値は文字列として指定し空白を含む場合には二重引用符"で囲む。値の比較

は文字列として行う。また二重引用符内に二重引用符を記号として指定する場合には、記号を二つ重ねて表記する。先頭第 1 カラムがアスタリスク (\*) の行はコメントとみなされる。

また、指定開始カラム 1 条件演算子 指定開始カラム 2 文字列長 (整数) の 4 つの項目を指定することにより、レコード中の 2 箇所のフィールドの比較を行える。条件演算子は&==, &!=, &>=, &<= (または &=<), &>, &<の 6 種を指定できる。

## 標準入出力

- (1) 標準入力  
固定長フォーマットを持つテキストデータを入力とする。
- (2) 標準出力  
選択されたレコードが出力される。
- (3) 標準エラー出力  
エラー情報と、入出力件数が報告される。

## 利用例

```
fixedAnd testm1.sel < testm1.dat > testm1sel.dat 2> testm1sel.log
          (1)           (2)           (3)           (4)
```

- (1) testm1.sel 選択条件ファイル

```
-----サンプルの開始-----
*testm1.sel
1 =< 05
3 != aA
-----サンプルの末尾-----
```

- (2) testm1.dat 標準入力

```
-----サンプルの開始-----
01aA
02bB
03cC
04dD
05eE
06fF
07gG
08hH
09iI
10jJ
-----サンプルの末尾-----
```

- (3) testm1sel.dat 標準出力

```
-----サンプルの開始-----
02bB
03cC
04dD
05eE
-----サンプルの末尾-----
```

- (4) testm1sel.log 標準エラー出力

```

-----サンプルの開始-----
*Controlfile testm1.sel
*Condition      Column Length Op Val
      1          1      2 =< 05
      2          3      2 != aA
*InputRecords   10
*SelectedRecords 4
*SkippedRecords 6
-----サンプルの末尾-----

```

## 2.6 fixedAndC

### 利用法

fixedAndC 指定開始カラム 比較演算子 値

### 機能

コマンドライン引数によってレコード選択の条件を指定し、標準入力したレコードのうち、条件を満たしたレコードを選択し、標準出力に出力する。選択のための条件は1つ演算によるもののみ指定できる。

### コマンドライン引数

- (1) 指定開始カラム (整数)  
選択のため比較判断の対象となる開始カラムを指定する。
- (2) 比較演算子  
次の6種の演算子が指定できる。  
eq (等しい), ne (等しくない), ge (データのカラムが第3引数以上), le (カラムが第3引数以下), gt (カラムが第3引数より大), lt (カラムが第3引数より小)。文字列としての大小によって比較を行う。(不等号はリダイレクションと混同するため使用を避けた。)
- (3) 比較文字列  
値に空白や不等号を含まない場合には、そのまま指定する。空白や不等号を含む場合には二重引用符”で囲む。

### 標準入出力

- (1) 標準入力  
固定長フォーマットを持つテキストデータを入力とする。
- (2) 標準出力  
選択されたレコードが出力される。
- (3) 標準エラー出力  
エラー情報と、入出力件数が報告される。

### 利用例

```
fixedAndC 1 1e 05 < testm1.dat > testm1selC.dat 2> testm1selC.log
          (1) (2) (3)          (4)          (5)          (6)
```

- (1) 1 指定開始カラム
- (2) 1e 比較演算子
- (3) 05 比較文字列
- (4) testm1.dat 標準入力 (内容は省略, fixedAnd の項で既出)
- (5) testm1selC.dat 標準出力

```
-----サンプルの開始-----
01aA
02bB
03cC
04dD
05eE
-----サンプルの末尾-----
```

- (6) testm1selC.log 標準エラー出力

```
-----サンプルの開始-----
*Condition      Column Length Op Val
          1          1          2 1e 05
*InputRecords      10
*SelectedRecords   5
*SkippedRecords    5
-----サンプルの末尾-----
```

## 2.7 fixedKeyDel/fixedKeySel

### 利用法

```
fixedKeyDel キーファイル名 キー開始カラム カラム長 標準入力開始カラム
fixedKeySel キーファイル名 キー開始カラム カラム長 標準入力開始カラム
```

### 機能

キーファイル中の指定されたフィールドに含まれる値の集合を求め、標準入力の指定カラムの値がこの集合に含まれる場合にレコードを削除 (fixedKeyDel)、または選択 (fixedKeySel) する。

### コマンドライン引数

- (1) キーファイル名  
キー値を含むファイル名を指定する。
- (2) キー開始カラム  
キーファイル中の指定フィールドの開始カラム位置を 1 以上の整数で指定する。キーファイル中のキーの値は重複していても構わない。
- (3) カラム長  
キーの長さ (バイト) を 1 以上の整数で指定する。

- (4) 標準入力開始コラム  
標準入力からのレコードにおいて、キーの値を比較する箇所の開始コラムを 1 以上の整数で指定する。

#### 標準入出力

- (1) 標準入力  
固定フォーマットのテキストレコード。行末には改行コードが必要。
- (2) 標準出力  
指定されたキーを含まないレコード (fixedKeyDel) , または指定されたキーを含むレコード (fixedKeySel)。
- (3) 標準エラー出力  
キーファイル名、キー開始コラム、コラム長、標準入力開始コラム、入力キー件数、ユニークなキーの件数、入力レコード件数、および出力レコード件数を表示する。

#### 利用例

```
fixedKeyDel keys.dat 2 2 4 < data1.dat > KeyDelout.dat 2> KeyDel.log
                (1)      (2) (3) (4) (5)                (6)                (7)
```

- (1) keys.dat キーファイル名

```
-----サンプルの開始-----
x33
x44
x55
-----サンプルの末尾-----
```

- (2) 2 キー開始コラム

- (3) 2 コラム長

- (4) 4 標準入力開始コラム

- (5) data1.dat 標準入力

```
-----サンプルの開始-----
1aa1111
2bb2222
3cc3333
4dd4444
5ee5555
6ff6666
7gg7777
-----サンプルの末尾-----
```

- (6) KeyDelout.dat 標準出力

```
-----サンプルの開始-----
1aa1111
2bb2222
6ff6666
7gg7777
-----サンプルの末尾-----
```

- (7) KeyDel.log 標準エラー出力

```

-----サンプルの開始-----
*Select or Delete by Keys
*KeyFile      keys.dat
*KeyStartCol  2
*Length       2
*StdinStartCol 4
*InputKeys    3
*UniqueKeys   3
*InputRecords 7
*OutputRecords 4
-----サンプルの末尾-----

```

## 2.8 fixedFreqs

### 利用法

fixedFreqs 指定開始カラム カラム長 [分離コードの指定]

### 機能

コマンドライン引数によって指定されたカラムに現れる文字列の頻度を求める。

### コマンドライン引数

- (1) 指定開始カラム  
頻度を求める範囲（フィールド）の開始カラムを 1 以上の整数で指定する。
- (2) カラム長フィールドの長さ（カラム長）を指定する。
- (3) 分離コードの指定出力において項目を区切る記号の指定。s ならば空白、t ならばタブ記号、c ならばカンマとなる。

### 標準入出力

- (1) 標準入力  
固定長のテキストファイル。行末には改行コードが必要。
- (2) 標準出力  
各行に指定フィールドの値、頻度、相対頻度が出力される。各行における数は、指定された分離コードによって区切られる。分離コードを空白に指定した場合は、頻度の出力は 12 カラム幅の右寄せで表示される。それ以外の場合には、分離コードに続いて直ちに数値が表示される。
- (3) 標準エラー出力  
開始カラム、フィールドの長さ、分離コードの指定値、入力レコード件数が表示される。

### 利用例

```

fixedFreqs 1 1 s < test1.dat > test1freqs.dat 2> test1freqs.log
           (1) (2) (3)      (4)      (5)                (6)

```

- (1) 1 指定開始カラム

- (2) 1 カラム長
- (3) s 分離コードの指定。s は空白の指定。省略時は空白。
- (4) test1.dat 標準入力 (内容は省略)
- (5) test1freqs.dat 標準出力

```

-----サンプルの開始-----
0          10 0.100000
1           9 0.090000
2          13 0.130000
3           8 0.080000
4           9 0.090000
5          10 0.100000
6          17 0.170000
7           5 0.050000
8          11 0.110000
9           8 0.080000
-----サンプルの末尾-----

```

- (6) test1freqs.log 標準エラー出力

```

-----サンプルの開始-----
*Values,Frequencies, and Ratios
*StartCol 1
*Length 1
*Separator s
*Total 100
-----サンプルの末尾-----

```

## 2.9 fixedLfreqs

### 利用法

fixedLfreqs 指定開始カラム 1 カラム長 1 指定開始カラム 2 カラム長 2 [分離コードの指定]

### 機能

コマンドライン引数によって指定された 2 箇所のカラムにあらわれる文字列の対の頻度を求める。

### コマンドライン引数

- (1) 指定開始カラム 1  
層別のためのフィールド (第 1 フィールド) の開始カラムを 1 以上の整数で指定する。
- (2) カラム長 1  
第 1 フィールドの長さ (カラム長) を指定する。
- (3) 指定開始カラム 2  
集計対象となるフィールド (第 2 フィールド) の開始カラムを 1 以上の整数で指定する。
- (4) カラム長 2  
第 2 フィールドの長さ (カラム長) を指定する。

(5) 分離コードの指定

出力において項目を区切る記号の指定。s ならば空白、t ならばタブ記号、c ならばカンマとなる。

標準入出力

(1) 標準入力

固定長のテキストファイル。行末には改行コードが必要。

(2) 標準出力

各行に第 1 フィールドの値、第 2 フィールドの値、および頻度が出力される。各行における数は、指定された分離コードによって区切られる。分離コードを空白に指定した場合は、頻度の出力は 12 カラム幅の右寄せで表示される。それ以外の場合には、分離コードに続いて直ちに数値が表示される。

(3) 標準エラー出力

第 1 フィールドの開始カラム、フィールドの長さ、第 2 フィールドの開始カラム、フィールドの長さ、分離コードの指定値、および入力レコード件数が表示される。

利用例

```
fixedLfreqs 1 1 2 1 s < test1.dat > test1Lfreqs.dat 2> test1Lfreqs.log
              (1) (2) (3) (4) (5)      (6)              (7)              (8)
```

(1) 1 指定開始カラム 1

(2) 1 カラム長 1

(3) 2 指定開始カラム 2

(4) 1 カラム長 2

(5) s 分離コードの指定。s は空白の指定。省略時は空白。

(6) test1.dat 標準入力 (内容は省略)

(7) test1Lfreqs.dat 標準出力

-----サンプルの開始-----

```
0 0          1
0 3          2
0 4          1
0 5          2
0 7          4
1 0          1
1 1          3
1 5          2
1 7          1
1 9          2
```

...

-----サンプルの末尾-----

(8) test1Lfreqs.log 標準エラー出力

-----サンプルの開始-----

```
*Layers, Values and Frequencies
*LayerCol 1
*LayerLen 1
*StartCol 2
*Length 1
*Separator s
*Total 100
```

-----サンプルの末尾-----

## 2.10 fixedDupl

### 利用法

fixedDupl

機能標準入力に指定された整列済みのファイルにおいて、重複するレコードが存在するかをテストする。

### コマンドライン引数

なし

### 標準入出力

- (1) 標準入力  
テキストデータを入力する。
- (2) 標準出力  
重複したレコードの行番号と内容を出力する。
- (3) 標準エラー出力  
入力レコード数と重複の件数を出力する。

### 利用例

```
sort test1withDupl.dat | fixedDupl > fixedDuplout.dat 2> fixedDuplout.log  
(1) (2) (3) (4) (5)
```

- (1) sort ソートコマンド。入力ファイルのレコードを整列する。
- (2) test1withDupl.dat 入力ファイル
- (3) fixedDupl 重複レコードの検出
- (4) fixedDuplout.dat 標準出力

```
-----サンプルの開始-----  
RecordNo:DuplicatedRecord  
5:0407934712  
6:0407934712  
11:0760443210  
12:0760443210  
85:7823440052  
88:8115555394  
103:9544139358  
-----サンプルの末尾-----
```

整列済みのファイルの第 5 レコードと第 6 レコードが第 4 レコードと重複している  
ので出力されている。また、同様に第 11 レコードと第 12 レコードは第 10 レコー  
ドと重複している。

第 85、88、103 レコードは各々それらの前の行と重複している。

- (5) fixedDuplout.log 標準エラー出力

```
-----サンプルの開始-----  
*InputRecords 107  
*Duplication 7  
-----サンプルの末尾-----
```

入力レコードは 107 である。上の標準出力に示されているように 7 行が重複してい  
るためにユニークなレコードは 100 件存在することがわかる。

## 2.11 fixedIns

### 利用法

`fixedIns` 開始カラム 挿入文字列

### 機能

標準入力から入力された各レコードの指定された箇所に、挿入文字列を挿入する。開始カラムにゼロを指定すると、行の先頭に挿入文字列を挿入する。( `fixedEdit` で利用するカラム情報ファイルの編集機能はない。)

### コマンドライン引数

#### (1) 開始カラム

文字列を挿入するカラム位置 (バイト) を 0 以上の整数で指定する。ゼロは行頭、1 は先頭バイトの直後を意味する。

#### (2) 挿入文字列

挿入すべき文字列を指定する。空白を含む文字列を指定する場合にはダブルクォートで囲む。

### 標準入出力

#### (1) 標準入力

固定長のテキスト。

#### (2) 標準出力

指定カラムの直後に、引数に指定した挿入文字列が挿入されたレコードが出力される。

#### (3) 標準エラー出力

指定された引数と入力レコード件数が表示される。

## 2.12 asctime

### 利用法

`asctime`

### 機能

現在の日時 (年月日、時分秒) を表示する。処理時間の見積もりに利用できる。

### コマンドライン引数

なし

### 標準入出力

#### (1) 標準入力

用いない。

#### (2) 標準出力

日時を表示する

- (3) 標準エラー出力  
用いない。

利用例

```
c:\>asctime  
Date and Time: Thu Jun 26 10:57:08 2003
```

## 2.13 prproof/prproof100

利用法

```
prproof [開始行 [終了行 [増分] ] ]  
または  
prproof100 [開始行 [終了行 [増分] ] ]
```

機能

表示すべきテキストファイルを標準入力から読み込み、標準出力に 70 バイト分 (prproof100 では 100 バイト分) を一行に表示し、それより多い部分は改行して表示する。表示を行う範囲は、コマンドライン引数で指定する。

コマンドライン引数

- (1) 開始行  
表示を開始する行を整数で指定する。コマンドライン引数が存在しない場合には、1 が仮定される。
- (2) 終了行  
表示を終了する行を整数で指定する。コマンドライン引数が存在しない場合、または 1 つのみ指定された場合には、4,294,967,295 (符号なし整数の最大) が仮定される。
- (3) 増分  
表示を行う間隔を指定する。1 ならばすべての行を表示し、2 であれば 1 行おきに表示する。コマンドライン引数の個数が 2 つ以下の場合には、1 が仮定される。

標準入出力

- (1) 標準入力  
Windows 上で作成されシフト JIS で記述されたテキストファイルを仮定する。現在のところ 1 行の最大バイト数は 4095 を仮定している。最終行の行末に改行コードが含まれていない場合には、最後の改行以降ファイルの終わりまでを 1 行とみなす。
- (2) 標準出力  
表示用に編集された形式で、指定された範囲の行が出力される。先頭には、カラムスケールが表示される。表示すべきデータの右端がシフト JIS コードで記述された 2 バイト文字の先頭バイトである場合には、その次のバイトを続けて同一行に表示し、次の行の行番号の 1 の位の位置から表示部分先頭にかけて-->を表示する。

### (3) 標準エラー出力

指定された表示範囲，および表示したレコードの件数が出力される．

利用例 Windows のコマンドウィンドウで次のように指定する．

```
prproof < sample.txt > sampleout.txt 2> sample.log
          (1)           (2)           (3)
```

(1) は標準入力へのファイルの指定をあらわす．(2) は標準出力．(3) は標準エラー出力の指定である（不等号のまえに数字の 2 を離さずに記述すると，標準エラー出力のリダイレクションの指定となる）

次は (2) の表示例である．先頭の日付と時刻は，表示を行った日時をあらわす．ここでの表示カラムは  $\text{T}_{\text{E}}\text{X}$  のフォントサイズの関係で，正しい位置にはなっていないが，PC 画面上の Notepad などにおいて等幅フォントを用いると，正しいカラム位置（2 バイト文字が 2 カラムちょうどを占める）に表示がされる．

```
Prproof: Sun Jun 19 00:26:31 2005
RecNo.  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7
  1 これは日本語のサンプルである．漢字やひらかなかたかななどがたくさんかいてあるし，英数字たとえば 012abcd などいろいろかいてあるのだ．これも日本語 -->のサンプルである．漢字やひらかなかたかななどがたくさんかいてあるし，英 -->数字たとえば 012abcd などいろいろかいてあるのだ．
  2 This is the second line. 000001111122222333334444455555666667777788888
    99999 第 2 行の終わりの日本語
  3 第 3 行目：日本語の表現と内容
  4 The fourth line
```

標準エラー出力は次のようになる．

```
*Start      1
*End        4294967295
*Increment  1
*PrintedLines 4
```

開始行などを指定する場合には，次のように変更する．

```
prproof 2 10 2 < sample.txt > sampleout.txt 2> sample.log
```

開始行を 2 とし，終了行を 10（この場合はファイルが 10 行未満なので，指定しても実質的な効果はない），増分を 2 とする．標準出力への表示は次のようになる．

```
Prproof: Sun Jun 19 00:50:05 2005
RecNo.  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7
  2 This is the second line. 000001111122222333334444455555666667777788888
    99999 第 2 行の終わりの日本語
  4 The fourth line
```

また、標準エラー出力は次のようになる。

```
*Start          2
*End            10
*Increment      2
*PrintedLines   2
```

## 2.14 fixedFldfreqs2

### 利用法

fixedFldfreqs2 入力カラム情報ファイル名 [カラム長の上限]

### 機能

入力カラム情報ファイル (fixedEdit と同様のもの) に指定された各カラムの頻度情報を求める。fixedFreqs によって得られる頻度集計を複数の指定カラムについて同時に求める。

### コマンドライン引数

- (1) 入力カラム情報ファイル  
fixedEdit の指定に用いたのと同様の、開始カラム、カラム長、変数名 (フィールド名) の 3 つを空白で区切って各行に記述したファイル。開始カラムがゼロ以下の数値を指定した場合には無視される。
- (2) カラム長の上限 (整数)  
この数値を超えるカラム長の指定の変数 (フィールド) は、集計の対象とならない。ゼロを指定すると制約を課さない (デフォルト)。

### 標準入出力

- (1) 標準入力  
固定長のテキストファイル。fixedEdit の入力と同様のもの。
- (2) 標準出力  
各変数 (フィールド) に現れる文字列の頻度、および相対頻度が表示される。
- (3) 標準エラー出力  
入力カラム情報ファイル名、指定されたカラム (フィールド) の数、集計結果の分離文字、レコードの総件数、カラム長上限 (無指定の場合はゼロ) を表示する。

利用例 fixedFldfreqs2 sample1.clm < testfoo.dat > testFldfreqs2.txt 2> testFldfreqs2.log  
(1) (2) (3) (4)

- (1) sample1.clm 入力カラム情報ファイル

```
-----サンプルの開始-----
*
 1 2 Col1-2
-1 1 Comma
 4 1 Col4
-----サンプルの末尾-----
```

(2) testfoo.dat 入力データ

```
-----サンプルの開始-----  
abcde1  
fghij2  
klmno3  
pqrst4  
uvwxy5  
abcde6  
fghij7  
klmno8  
pqrst9  
uvwxy0  
-----サンプルの末尾-----
```

(3) testFldfreqs2.txt 出力ファイル (標準出力)  
カラム文字列、頻度 (12桁)、相対頻度が空白で区切られて表示される。

```
-----サンプルの開始-----  
*---+----+No.1 Column 1 Length 2 Col1-2  
ab          2 0.200000  
fg          2 0.200000  
kl          2 0.200000  
pq          2 0.200000  
uv          2 0.200000  
*---+----+No.2 Column 4 Length 1 Col4  
d           2 0.200000  
i           2 0.200000  
n           2 0.200000  
s           2 0.200000  
x           2 0.200000  
-----サンプルの末尾-----
```

(4) testFldfreqs2.log 実行メッセージ (標準エラー出力)

```
-----サンプルの開始-----  
Line 2 Column -1 Legth 1 was ignored.  
*Values,Frequencies, and Ratios  
*InputColumnInfo  sample1.clm  
*Nfields          2  
*Separator        s  
*Total            10  
*LengthThreshold  0  
-----サンプルの末尾-----
```

## 2.15 fixedFldlfreqs2

### 利用法

fixedFldlfreqs2 カラム対情報ファイル名 カラム開始位置 1 カラム開始位置 2

### 機能

入力カラム対情報ファイルに指定されたカラム対のクロス頻度情報を求める。fixedLfreqsによって得られる層別頻度集計と同等の結果を、複数の指定カラム対について同時に求める。集計対象となるデータ (成績ファイル) は、標準入力に指定する。集計結果は、標準出力に表示する。指定パラメータなどについては、標準エラー出力に表示する。

### コマンドライン引数

(1) カラム対情報ファイル名

集計対象となるファイルのカラム対の情報を各行に指定する。カラム対情報ファイルの各行には、

開始カラム1 カラム長1 開始カラム2 カラム長2 項目名

の4つの項目を記述する。開始カラム1、カラム長1、開始カラム2、カラム長2は正の整数を指定する。項目名は任意の文字列。開始カラム2とカラム長2によって指定される文字列の値によって層別が行われる。開始カラム1とカラム長1によって指定された文字列の層別頻度が、標準出力に表示される。指定パラメータについての情報、入出力件数などが標準エラー出力に表示される。先頭カラムがASCIIのアスタリスク\*である行は、コメントとなる。

(2) カラム開始位置1

「開始カラム1」と「カラム長1」によって指定されるカラムの基準となる位置を示す。レコードの先頭カラムを1とすると、「カラム開始位置1」+「開始カラム1」-1が集計対象の開始カラムとなる。

(3) カラム開始位置2

「開始カラム2」と「カラム長2」によって指定されるカラムの基準となる位置を示す。レコードの先頭カラムを1とすると、「カラム開始位置2」+「開始カラム2」-1が層別キーの開始カラムとなる。

標準入出力

(1) 標準入力

集計対象となる固定長フォーマットのデータ(テキスト)

(2) 標準出力

指定されたカラム対にあらわれる文字列の対の頻度。「層別キー」、「集計対象」、「頻度」の順に表示される。(パラメータ指定と「層別キー」、「集計対象」の順が逆に表示される)

(3) 標準エラー出力指定パラメータおよび処理件数についての報告

利用例

```
fixedAndC 31 eq J1 < H180MKakutei.data | fixedFldlfreqs2 test2input.clm2  
          (1)                                     (2)
```

```
(続き) 55 305 > test2out.log 2> test2err.log  
        (3) (4)          (5)          (6)
```

(1) fixedAndC 31 eq J1 < H180MKakutei.data 標準入力の作成。指定科目のレコード(31-2カラムが'J1')のものを抽出する例。

(2) test2input.clm2 カラム対情報ファイル名を指定する。以下は指定の例。各行は集計対象カラムの開始位置、長さ、層別カラムの開始位置、長さ、項目名の順に記述する。この例では集計対象カラムは解答であり、層別カラムは項目別の得点であり(文字として扱う)、項目名は設問の識別記号である。

```
----- サンプルの開始 -----  
*2006 J1 Honshiken  
* Ans Len Score ScLen Name  
1 1 1 2 2006-J1-H-01-A-01_1
```

```

2 1 3 2 2006-J1-H-01-A-02_2
3 1 5 2 2006-J1-H-01-B-01_3
4 1 7 2 2006-J1-H-01-B-02_4
5 1 9 2 2006-J1-H-01-B-03_5
6 1 11 2 2006-J1-H-01-B-04_6
...
...
...
46 1 91 2 2006-J1-H-06-A-04_46
47 1 93 2 2006-J1-H-06-A-05_47
48 3 99 2 2006-J1-H-06-B-99_48
----- サンプルの終わり -----

```

- (3) カラム開始位置 1。集計対象カラムの基準位置を示す。第 55 カラムが、解答データ（設問への解答部分）の先頭であることを指定する。
- (4) カラム開始位置 2。層別カラムの基準位置を示す。設問別のスコア（設問別個別得点）が記録されている先頭カラムを指定する。
- (5) test2outlog.txt 出力ファイル（標準出力）各設問ごとの先頭に '\*----+----+No' で始まる見出しが出力され、次いで各行に層別コード（得点）、空白、集計対象コード（解答）、空白、頻度（カラム数 12）が出力される。ここで示した例では、最初の層別コードは 00 または 02 であり、解答が 2 のときのみ得点が 02 であり、他は 00 となっている。最初の項目で得点を得たものが 253 名いることがわかる。

```

----- サンプルの開始 -----
*----+----+No 1 Column: 55 Length: 1 Layer: 305 LayerLen: 2
(続き) Field: 2006-J1-H-01-A-01_1
00          1
00 1        14
00 3         6
00 4        74
02 2        253
*----+----+No 2 Column: 56 Length: 1 Layer: 307 LayerLen: 2
(続き) Field: 2006-J1-H-01-A-02_2
00          1
00 1         3
00 2        54
00 3        25
02 4        265
*----+----+No 3 Column: 57 Length: 1 Layer: 309 LayerLen: 2
(続き) Field: 2006-J1-H-01-B-01_3
...
----- サンプルの終わり -----

```

- (6) test2errlog.txt 実行メッセージ（標準エラー出力）Separator は、標準出力における区切りの指定。s は空白、t はタブ、c はコンマをあらわすが、現状では空白

のみ。Nfields は指定された項目の数。AnsTopColumn は集計対象カラムの基準位置、ScoreStartColumn は層別カラムの基準位置を示す。

```

----- サンプルの開始 -----
*Values, Layers, and Frequencies
*InputColumnInfo test2input.clm2
*Nfields          44
*Separator        s
*Total            348
*AnsTopColumn     55
*ScoreStartColumn 305
----- サンプルの終わり -----

```

### 3 処理時間

PC でデータ処理を行った場合の経過時間の例を表 2 示す。データがメモリキャッシュに取り込まれている場合には、処理速度が大きく改善するので、処理時間は実際にはかなり変動する。

用いた PC の機種、周辺機器等は表 3 に示した。

表 2: 処理時間

プログラム	データ レコード長	データ 件数	経過時間 (単位秒)	備考
(1)	3900	33 万件	29	3 バイト幅 x1 の頻度集計 (頻度パターン 180 種)
(2)	3900	33 万件	10	選択件数 28 件

(1) fixedFreqs 1 3 < DATA > UTF1

(2) fixedAndC 1 eq 010 < DATA > UTF2

表 3: PC 周辺機器等仕様

機種	HP xw4400
CPU	Intel Core2 Duo 6600 2.40GHz
メモリ	4.0GB (有効 3.25GB)
HDD	Intel 82801 GR/GH SATA RAID (RAID1)/ ST3250624AS x2
OS	WindowsXP Professional (SP3)
C コンパイラ	Intel C++ ver10.1
コンパイラオプション	/O2

## 4 プログラムの所在

ソースプログラムおよび、Windows2000/XP 用の実行ファイルを  
<http://www.rd.dnc.ac.jp/~otsu/>で公開する。

### 参考文献

- [1] Intel (2007). *Intel C++ compiler documentation*.  
<http://developer.intel.com/software/products/>
- [2] Harbison,S.P. & Steele Jr.,G.L. (1991). *C: A Reference Manual 3rd ed.*, Prentice-Hall. ( 齋藤信夫監訳 1994, 新・詳説C言語, ソフトバンク).
- [3] MinGW ホームサイト <http://www.mingw.org/>
- [4] Stroustrup,B. (2000). *The C++ programming language special edition*, Addison-Wesley.