別添資料

以下、本統括・分担研究報告書全体に係る別添資料を示す。
順に、以下の構成となっている。

- Iris 用入力データ生成プログラムソース（主要な部分抜粋）
  - メインプログラム（parse.pl）
  - 処理用ライブラリソース
    - GKAI::Coding.pm
    - GKAI::Kikan.pm
    - GKAI::Util.pm
  - Iris 入力データ作成用 日本独自 ICD コード変換ルール
  - Iris 入力データ作成用 期間表現正規化ルール

- 機械学習による原死因変更有無予測結果（一部抜粋）

[メインプログラム]

```perl
#! /usr/bin/env perl

#=====================================================================#
#==                                                                 ==#
#==                          [ parse.pl ]                           ==#
#==                                                                 ==#
#==                                                                 ==#
#=====================================================================#

use warnings;
use v5.28;

use utf8;
use Encode;
use FindBin;
use lib "$FindBin::Bin/lib";
use GKAI::Util;
use GKAI::Coding;
use GKAI::Kikan;

binmode STDIN,  ":utf8";
binmode STDOUT, ":utf8";

if ( $ARGV{HELP} ) {
    &printHelp();
}

#== 初期設定 ==========================================================
my $switch = "";    # スイッチ
my $kikanrule = decode_utf8($FindBin::Bin) . '/rules/期間なおし_ver2.csv';
my $tikanrule = decode_utf8($FindBin::Bin) . '/rules/Iris用ICD変換.csv';
my $sdm = decode_utf8($FindBin::Bin) . '/rules/504-han-withICD.txt';
my $uncode = decode_utf8($FindBin::Bin) . '/rules/未code化傷病名用リスト.txt';
my $normalizekanji = decode_utf8($FindBin::Bin) . '/rules/記載変換ルール.txt';
my $dup = {};
my $gu = new GKAI::Util();
my $gc = new GKAI::Coding('SDM' => $sdm, 'ICDtikan' => $tikanrule,'UNCODE' =>
$uncode, 'ICDtikan2' => $normalizekanji);
my $gk = new GKAI::Kikan('kikanRule' => $kikanrule);


#== 処理本体 ==========================================================

#== ALL
open IDENTALL, ">:utf8", "ALL/Ident.txt";
open MEDCODALL, ">:utf8", "ALL/MedCod.txt";

#== 5分割
open my $OUTI1, ">:utf8", "5分割/Ident1.txt";
open my $OUTI2, ">:utf8", "5分割/Ident2.txt";
open my $OUTI3, ">:utf8", "5分割/Ident3.txt";
open my $OUTI4, ">:utf8", "5分割/Ident4.txt";
open my $OUTI5, ">:utf8", "5分割/Ident5.txt";

open my $OUTM1, ">:utf8", "5分割/MedCod1.txt";
open my $OUTM2, ">:utf8", "5分割/MedCod2.txt";
open my $OUTM3, ">:utf8", "5分割/MedCod3.txt";
open my $OUTM4, ">:utf8", "5分割/MedCod4.txt";
open my $OUTM5, ">:utf8", "5分割/MedCod5.txt";

#== テスト用（10件ごと抽出）
open TESTI, ">:utf8", "Test/Ident.txt";
open TESTM, ">:utf8", "Test/MedCod.txt";

#== テスト用（500件ごと抽出）
open TESTIS, ">:utf8", "Test2/Ident.txt";
open TESTMS, ">:utf8", "Test2/MedCod.txt";

#== IRIS にかけたリスト
open IRIS, ">:utf8", "ALL/ForIris.txt";
open IRIS5, ">:utf8", "5分割/ForIris.txt";
open IRIST, ">:utf8", "Test/ForIris.txt"; # テスト用
open IRISTS, ">:utf8", "Test2/ForIris.txt"; # テスト用

#== 期間のパターン
open KIKAN, ">:utf8", "期間表現/Kikan_match_all.txt";

open IN, "<:utf8", "../01_NewShiboJoin/new_shibo_join.tsv";
my $c = 0;
my $c2 = 0;
while (my $str = <IN>) {
    $c ++;

    my $o1 = $OUTI1;
    my $o2 = $OUTM1;
    if ($c <= 1000000){ $o1 = $OUTI1; $o2 = $OUTM1 }
    elsif ($c <= 2000000){ $o1 = $OUTI2; $o2 = $OUTM2}
    elsif ($c <= 3000000){ $o1 = $OUTI3; $o2 = $OUTM3}
    elsif ($c <= 4000000){ $o1 = $OUTI4; $o2 = $OUTM4}
    elsif ($c <= 5500000){ $o1 = $OUTI5; $o2 = $OUTM5}

    if ($c != 0 ) {
        my @a = split(/\t/, $str);

      # 病名は 10,12,14,16,18
      my $d = $gc->ICDCoding($a[10], $a[12], $a[14], $a[16], $a[18]);

      # 期間は 11,13,15,17,19
        my @kikan = ();
      my $k1 = $gk->getKikan($a[11]);
      my $k2 = $gk->getKikan($a[13]);
        my $k3 = $gk->getKikan($a[15]);
        my $k4 = $gk->getKikan($a[17]);
        my $k5 = $gk->getKikan($a[19]);
        @kikan = ($k1, $k2, $k3, $k4, $k5);

        #-- 期間のDEBUG用 ------------
        my $dk1 = $gk->debug($a[11]);
        my $dk2 = $gk->debug($a[13]);
        my $dk3 = $gk->debug($a[15]);
        my $dk4 = $gk->debug($a[17]);
        my $dk5 = $gk->debug($a[19]);
        if ($dk1 ne '') {say KIKAN $dk1}
        if ($dk2 ne '') {say KIKAN $dk2}
        if ($dk3 ne '') {say KIKAN $dk3}
        if ($dk4 ne '') {say KIKAN $dk4}
        if ($dk5 ne '') {say KIKAN $dk5}

        # 全てにICDコードが当たった
        if ($d->{HIT}){

            my $key = $gu->keta($a[0]);

            # Identテーブル
            my @identArr = ();
            for (my $i = 0; $i < 44; $i++){ $identArr[$i] = '' };
            $identArr[0] = $key;
            $identArr[2] = $gu->ymd($a[47]); # DateBirth (birth_ymd)
            $identArr[3] = $gu->ymd($a[50]); # DateDeath (death_ymd)
            $identArr[5] = $a[45]; # Sex (sex)
            $identArr[6] = '0';    # MannerOfDeath ('0')
            $identArr[9] = 'Initial';  # Status ('Initial')
            $identArr[10] = 'No'; # Reject ('No')
            $identArr[11] = 'Automatic'; # Coding ('Automatic')
            my $identRes = join("\t", @identArr);

            # ALL
            print IDENTALL "$identRes\r\n";
            # 5分割
            print $o1 "$identRes\r\n";
            # テスト
            if ($c % 10 == 0){ print TESTI "$identRes\r\n" }
            if ($c % 500 == 0){ print TESTIS "$identRes\r\n" }

            # MedCodテーブル
            for (my $i = 0; $i <= 4; $i++){
                my @medcodArr = ();
                for (my $i = 0; $i < 7; $i++){ $medcodArr[$i] = '' };
                if($d->{DIS}[$i] ne ''){
                    my $icd = $d->{DIS}[$i];
                    # 期間がある場合付け加える。
                    if ($kikan[$i] ne '' and $kikan[$i] ne 'N/A'){$icd .= '('
. $kikan[$i] . ')'}
                    #----
                    my $medcodnum = 0;
                    if ($i <= 3) {$medcodnum = $i}
                    else {$medcodnum = 5}
                    #----
                    $medcodArr[0] = $key;
                    $medcodArr[1] = $medcodnum;
                    $medcodArr[2] = '';
                    $medcodArr[3] = $icd;
                    $medcodArr[4] = '';
                    $medcodArr[5] = 1;
                    $medcodArr[6] = 1;
                    my $medcodRes = join("\t" , @medcodArr);

                    # ALL
                    print MEDCODALL "$medcodRes\r\n";
                    # 5分割
                    print $o2 "$medcodRes\r\n";
                    # テスト
                    if ($c % 10 == 0){ print TESTM "$medcodRes\r\n" }
                    if ($c % 500 == 0){ print TESTMS "$medcodRes\r\n" }
                }
            }

            #== ForIris.txt にID記入
            # ALL
            say IRIS $a[0];
            # 5分割
            say IRIS5 $a[0];
            # テスト
            if ($c % 10 == 0){ say IRIST $a[0] }
            if ($c % 500 == 0){ say IRISTS $a[0] }
        }
        # INFO
        if ($c % 10000 == 0 ) { say STDERR "[INFO]: $c death certificates processed." }
    }
}
close(IN);


#== ALL
close(IDENTALL);
close(MEDCODALL);

#== 5分割
close($OUTI1);
close($OUTI2);
close($OUTI3);
close($OUTI4);
close($OUTI5);

close($OUTM1);
close($OUTM2);
close($OUTM3);
close($OUTM4);
close($OUTM5);

#== テスト用（10件ごと）
close(TESTI);
close(TESTM);

#== テスト用（5000件ごと）
close(TESTIS);
close(TESTMS);

#== IRIS にかけたリスト
close(IRIS);
close(IRIS5);
close(IRIST);
close(IRISTS);

#== 期間表現
close(KIKAN);
```

```perl
package GKAI::Coding;

#=====================================================================#
#==                                                                 ==#
#==                        [ GKAI::Coding ]                         ==#
#==                                                                 ==#
#==                                                                 ==#
#=====================================================================#

use warnings;
use v5.28;
use utf8;
use GKAI::Util;
use Encode;
use Carp qw(croak);
use Data::Dumper;

binmode STDIN,  ":utf8";
binmode STDOUT, ":utf8";
binmode STDERR, ":utf8";

#== 初期設定 =========================================================
our $VERSION = 1.00;
our %DEFAULT = ();

#== コンストラクタ ===================================================
sub new {
    my $class = shift;
    my $this  = {};
    bless $this, $class;

    #== 変数の設定 ==================
    my (%option) = @_;
    my $u = new GKAI::Util;
    $this->{UTIL} = $u;
    $this->{SDM} = {};
    $this->{TIKANRULE} = {};
    $this->{UNCODE} = {};
    $this->{TIKANRULE2} = {};

    #== 初期設定 ====================
    $this->loadSDM($option{"SDM"}, $option{"UNCODE"});
    $this->loadICDtikanRule($option{"ICDtikan"});
    $this->loadICDtikanRule2($option{"ICDtikan2"});
    return $this;
}

#== デストラクタ =====================================================
sub DESTROY {
    my $this = shift;
}

#== Methods ===========================================================

#----------------------------------------------------------------------
# [loadSDM]
#----------------------------------------------------------------------
sub loadSDM{
    my $this = shift;
    my ($in, $in2) = @_;
    open SDM, "<:utf8", $in or croak("[ERROR]: cannot load SDM");
    say STDERR '[INFO]: 標準病名マスター （' . $in . "） loaded";
    while (<SDM>){
      chomp;
      my ($d, $i) = split(/\t/);
        $this->{SDM}{$d} = $i;
    }
    close(SDM);

    open UNCODE, "<:utf8", $in2 or croak("[ERROR]: cannot load UNCODE");
    say STDERR '[INFO]: 未コード化傷病名用マスター （' . $in2 . "） loaded";
    while (<UNCODE>){
      chomp;
```

```
        my ($d, $i) = split(/\t/);
        $this->{SDM}{$d} = $i;
    }
    close(UNCODE);
}

#-----------------------------------------------------------------------
# [loadICDtikanRule]
#-----------------------------------------------------------------------
sub loadICDtikanRule{
    my $this = shift;
    my ($in) = @_;
    my $info = {};
    $info->{NUM} = 0;
    $info->{PATH} = $in;
    say STDERR "[INFO]: ICD10コード強制置換ルール　(" . $info->{PATH} . ") loading...";
    my $csv = Text::CSV::Encoded->new({
        encoding_in => 'UTF-8',
        encoding_out => 'UTF-8'
    });
    open my $IN, '<:utf8', $in or die "cannnot open $in $!";
    while (my $fields = $csv->getline($IN)){
        # 現在有効なルールのみ
        if ($fields->[4] =~ /^(1|2)$/){
            $info->{NUM} ++;
            my $p = $fields->[0];
            $this->{TIKANRULE}{$p}{FREQ} = $fields->[2];
            $this->{TIKANRULE}{$p}{NEW} = $fields->[1];
        }
    }
    close($IN);

    say STDERR "[INFO]: 有効ICD10コード強制置換ルール数　" . $info->{NUM};
}


sub loadICDtikanRule2{##異字体の正規化
    my $this = shift;
    my ($in) = @_;
    my $num = 0;
    open my $fh, '<:encoding(utf8)', $in;  # auto decoding on read
    while (my $line = <$fh>) {
        chomp ($line);
        Encode::_utf8_off($line);
        $line= decode('UTF-8', $line);
        my @d=split(/\t/, $line, -1);
        next if (length($d[0])<1 );
        $num++;
        $this->{TIKANRULE2}{$num}{before}=$d[0];
        $this->{TIKANRULE2}{$num}{after}=$d[1];
    }
    $this->{TIKANRULE2}{NUM}=$num;
    close($fh);
}


sub tikanICD2{
    my $this = shift;
    my ($in) = @_;
    for my $i(1..$this->{TIKANRULE2}{NUM})
    {
        my $before=$this->{TIKANRULE2}{$i}{before};
        my $after=$this->{TIKANRULE2}{$i}{after};
        $before=quotemeta $before;
        $after=quotemeta $after;
        if ($in =~ /$before/) {
            $in =~ s/$before/$after/g;
        }
    }
    return $in;
}

```

```perl

#---------------------------------------------------------------------
# [tikanICD]
#---------------------------------------------------------------------
sub tikanICD{
    my $this = shift;
    my ($in) = @_;
    my $ret = '';
    if (exists($this->{TIKANRULE}{$in})){
        $ret = $this->{TIKANRULE}{$in}{NEW};
    }
    else {
        $ret = $in;
    }
    return $ret;
}


#---------------------------------------------------------------------
# [byomei2icd]  病名から何らかのICDコードを返す。なければ空文字
#---------------------------------------------------------------------
sub byomei2icd{
    my $this = shift;
    my ($in) = @_;
    my $code = '';
    my $ret = '';

    my $sdm = $this->{SDM};

    #--- 仮のICDコードを引く
    # 標準病名 + 未コード化傷病名用セットを引く
    if (exists($sdm->{$in})){
      $code = $sdm->{$in};
    }

    # 部分一致の場合
    else{
      $code = ''; #★
    }

    #--- IRIS入力用にICDコードを無理矢理変更
    $ret = $this->tikanICD($code);
    return $ret;
}

#---------------------------------------------------------------------
# [ICDCoding]
#---------------------------------------------------------------------
#-----------------------
sub ICDCoding{
    my $this = shift;
    my @a = @_;
    my $ret = {};
    my $u = $this->{UTIL};
    my $sdm = $this->{SDM};
    $ret->{HIT} = 1;
    $ret->{DIS} = [];
    $ret->{UNKNOWN} = {};
    for (my $i = 0; $i < @a; $i++){
        my $byomei = $a[$i];
        $ret->{DIS}[$i] = '';

        # 前後の全角/半角空白の連続、「、」の前後の全角空白のみを除去
        my $tmp1 = $u->rmSpaceSimple($byomei);

        # カッコで囲まれた部分を除外する
        # 括弧を取る
        $tmp1 =~ s/<.*?>//g if( $tmp1 );
        $tmp1 =~ s/\(.*?\)//g if( $tmp1 );
        $tmp1 =~ s/（.*?）//g if( $tmp1 );
        $tmp1 =~ s/【.*?】//g if( $tmp1 );
        $tmp1 =~ s/〔.*?〕//g if( $tmp1 );

        # スタート文字の場合削除
        my @mark1=(">",'\)',"）","】","〕");

        foreach my $m (@mark1){
            $m=quotemeta $m;
            $tmp1=~s/^$m//g if( $tmp1 );
        }

        # 閉じ括弧がなければ全部削除
        my @mark2=("<",'(',"（","【","〔");

        foreach my $m (@mark2){
            $m=quotemeta $m;
            $tmp1 =~ s/$m.*$//g if( $tmp1 );
        }

        $tmp1 = $this->tikanICD2( $tmp1 );

        # 空文字列でなければ
        if ($tmp1 ne '') {

            my $res = ''; # ICD Coding 結果

            #--- パターン解析用
            #my $pattern = $tmp1;
            #$pattern =~ s/[^　、・]/x/g;
            #$pattern =~ s/x+/x/g;
            #say $pattern;

            #--- 複雑なパターン
            # 全角空白あり 読点なし
            if ($tmp1 =~ /　/ and $tmp1 !~ /、/) {

                my @b = split(/　+/, $tmp1);
                for (my $i = 0; $i < @b; $i++) {
                    my $tmp2 = $u->zen2han($b[$i]);
                    my $tmp3 = $this->byomei2icd($tmp2);
                    if ($tmp3 ne ''){
                        $res .= ','. $tmp3;
                }
                else {
                  $res .= ',★';
                }
                }
            }

            # 全角空白なし　読点あり
            elsif ($tmp1 !~ /　/ and $tmp1 =~ /、/) {
                #先頭と最後の読点は削除
                $tmp1 =~ s/^、//g;
                $tmp1 =~ s/、$//g;
                my @b = split(/、+/, $tmp1);
                for (my $i = 0; $i < @b; $i++) {
                    my $tmp2 = $u->zen2han($b[$i]);
                    my $tmp3 = $this->byomei2icd($tmp2);
                    if ($tmp3 ne ''){
                        $res .= ','. $tmp3;
                    }
                    else {
                        $res .= ',★';
                    }
                }
            }

            #--- 残りは"１病名"とみなす
            else {
                my $tmp2 = $u->zen2han($tmp1);
                my $tmp3 = $this->byomei2icd($tmp2);
                if ($tmp3 ne ''){
                    $res .= ','. $tmp3;
                }
                else {
                    $res .= ',★';
                }
                }
            }
            my $tmpres = substr($res, 1);

            # ★を含んだら HIT を 0にする
            if ($tmpres =~ /★/){
                $ret->{HIT} = 0;
            }
            $ret->{DIS}[$i] = $tmpres;
        }
    }
    return $ret;
}

1;


```

```perl
package GKAI::Kikan;

#=====================================================================#
#==                                                                 ==#
#==                         [ GKAI::Kikan ]                         ==#
#==                                                                 ==#
#==                                                                 ==#
#=====================================================================#

use warnings;
use v5.28;
use utf8;
use Encode;
use Carp qw(croak);
use Text::CSV::Encoded;
use GKAI::Util;

binmode STDIN,  ":utf8";
binmode STDOUT, ":utf8";
binmode STDERR, ":utf8";

#== 初期設定 ==========================================================
our $VERSION = 1.00;
our %DEFAULT = ();

#== コンストラクタ ====================================================
sub new {
    my $class = shift;
    my $this  = {};
    bless $this, $class;

    #== 変数の設定 ==================
    my (%option) = @_;
    $this->{KIKANRULE} = {};
    $this->{UTIL} = new GKAI::Util();

    #== 初期設定 ====================
    $this->loadKikanRule($option{"kikanRule"}); #

    #===============================
    return $this;
}

#== デストラクタ ======================================================
sub DESTROY {
    my $this = shift;
}

#== Methods ===========================================================

#---------------------------------------------------------------------
# [getKikan]
#---------------------------------------------------------------------
sub getKikan{
    my $this = shift;
    my ($in) = @_;
    my $u = $this->{UTIL};
    my $in2 = $u->rmSpaceSimple($in);
    my $str = $u->zen2han($in2);
    my $p = $this->getKikanPattern($in);
    my $ret = '';
    if ($str ne '') {
        if (exists($this->{KIKANRULE}{$p})){
            my $f = $this->{KIKANRULE}{$p}{FREQ};
            my $v = $this->{KIKANRULE}{$p}{VAR};
            my $v2 = $v;
            my $u = $this->{KIKANRULE}{$p}{UNIT};
            my $r = '';
            my @a = (); # 元の数値表現
            while ($str =~ /\d{1,}/g){ push(@a, $&) }
            my @b = (); # 正規化数値表現 ('09分' 等対策)
            for (my $i = 0; $i < @a; $i++){
                my $tmp = $a[$i] + 0;
                push(@b, $tmp);
            }
            if ($v2 =~ /\$1/){ $v2 =~ s/\$1/$b[0]/ }
            if ($v2 =~ /\$2/){ $v2 =~ s/\$2/$b[1]/ }
            if ($v2 =~ /\$3/){ $v2 =~ s/\$3/$b[2]/ }
            if ($v2 ne ''){ $r = eval($v2)};
            my $r2 = $r . $u;
            if ($r2 ne '' or $r2 ne 'N/A') {$ret = $r2}
        }
    }
    return $ret;
}

#-----------------------------------------------------------------------
# [debug]
#-----------------------------------------------------------------------
sub debug{
    my $this = shift;
    my ($in) = @_;
    my $u = $this->{UTIL};
    my $in2 = $u->rmSpaceSimple($in);
    my $str = $u->zen2han($in2);
    my $p = $this->getKikanPattern($in);
    my $ret = '';

    if ($str ne '') {
        if (exists($this->{KIKANRULE}{$p})){
            my $f = $this->{KIKANRULE}{$p}{FREQ};
            my $v = $this->{KIKANRULE}{$p}{VAR};
            my $v2 = $v;
            my $u = $this->{KIKANRULE}{$p}{UNIT};
            my $r = '';
            my @a = (); # 元の数値表現
            while ($str =~ /\d{1,}/g){ push(@a, $&) }
            my @b = (); # 正規化数値表現（'09分' 等対策）
            for (my $i = 0; $i < @a; $i++){
                my $tmp = $a[$i] + 0;
                push(@b, $tmp);
            }
            if ($v2 =~ /\$1/){ $v2 =~ s/\$1/$b[0]/ }
            if ($v2 =~ /\$2/){ $v2 =~ s/\$2/$b[1]/ }
            if ($v2 =~ /\$3/){ $v2 =~ s/\$3/$b[2]/ }
            if ($v2 ne ''){ $r = eval($v2)};
            my $r2 = $r . $u;
            $ret = $str . "\t" . $p . "|" . $f . "|" . $v . "|" . $u . "\t" .
$v2 . "|" . $r . "\t" . $r2;
        }
    }
    return $ret;
}

#-----------------------------------------------------------------------
# [loadKikanRule]
#-----------------------------------------------------------------------
sub loadKikanRule{
    my $this = shift;
    my ($in) = @_;
    my $info = {};
    $info->{NUM} = 0;
    $info->{PATH} = $in;
    say STDERR "[INFO]: 期間正規化ルール (" . $info->{PATH} . ") loading...";
    my $csv = Text::CSV::Encoded->new({
        encoding_in => 'UTF-8',
        encoding_out => 'UTF-8'
    });
    open my $IN, '<:utf8', $in or die "cannnot open $in $!";
    while (my $fields = $csv->getline($IN)){
        # 現在有効なルールのみ
        if ($fields->[5] eq '1'){
            $info->{NUM} ++;
            my $p = $fields->[1];
            $this->{KIKANRULE}{$p}{FREQ} = $fields->[0];
            $this->{KIKANRULE}{$p}{VAR} = $fields->[3];
            $this->{KIKANRULE}{$p}{UNIT} = $fields->[4];
        }
    }
    close($IN);

    say STDERR "[INFO]: 有効期間正規化ルール数  " . $info->{NUM};
}

#---------------------------------------------------------------------
# [getKikanPattern]
#---------------------------------------------------------------------
sub getKikanPattern{
    my $this = shift;
    my ($in) = @_;
    my $u = $this->{UTIL};
    my $in2 = $u->rmSpaceSimple($in);
    my $in3 = $u->zen2han($in2);
    $in3 =~ tr/0-9/xxxxxxxxxx/;
    return $in3;
}


1;


```

```perl
package GKAI::Util;

#=====================================================================#
#==                                                                 ==#
#==                         [ GKAI::Util ]                          ==#
#==                                                                 ==#
#==                                                                 ==#
#=====================================================================#

use warnings;
use v5.28;
use utf8;
use Encode;
use Carp qw(croak);
use Text::CSV::Encoded;

binmode STDIN,  ":utf8";
binmode STDOUT, ":utf8";
binmode STDERR, ":utf8";

#== 初期設定 =========================================================
our $VERSION = 1.00;
our %DEFAULT = ();

#== コンストラクタ ====================================================
sub new {
    my $class = shift;
    my $this  = {};
    bless $this, $class;

    #== 変数の設定 ==================
    my (%option) = @_;
    #== 初期設定 ====================

    return $this;
}

#== デストラクタ ======================================================
sub DESTROY {
    my $this = shift;
    #untie( %{ $this->{DATA} } );
}

#== Methods ===========================================================

#----------------------------------------------------------------------
# [ymd]  年月日のフォーマット合わせ
#    3020421  4280502
#----------------------------------------------------------------------
sub ymd{
    my $this = shift;
    my $in = shift or croak("[ERROR]: No Input");
    my $y = '';
    my $t1 = substr($in, 0, 1);
    my $t2 = substr($in, 1, 2);
    my $m = substr($in, 3, 2);
    my $d = substr($in, 5, 2);
    my $ret = '';

    # 明治
    if ($t1 eq '1'){
  $y = 1867 + $t2;
    }
    # 大正
    elsif ($t1 eq '2'){
  $y = 1911 + $t2;
    }
    # 昭和
    elsif ($t1 eq '3'){
  $y = 1925 + $t2;
    }
    # 平成
    elsif ($t1 eq '4'){
```

```
  $y = 1988 + $t2;
    }
    $ret = $y . '/' . $m . '/' . $d;
    if ($in =~ /ＶＶ/){
  $ret = '1940/01/01';
    }
    return $ret;
}

#---------------------------------------------------------------------
# [keta]  certificate key 用桁合わせ
#---------------------------------------------------------------------
sub keta{
    my $this = shift;
    my ($in) = @_;
    my $ret = '';
    $ret = "0" x (7 - length($in)) . $in;
    return $ret;
}

#---------------------------------------------------------------------
# [zen2han]  （英数記号全て）全角 => 半角変換
#---------------------------------------------------------------------
sub zen2han {
    my $this = shift;
    my ($str) = @_;
    $str =~ tr/０１２３４５６７８９/0-9/;
    $str =~ tr/ＡＢＣＤＥＦＧＨＩＪＫＬＭＮＯＰＱＲＳＴＵＶＷＸＹＺ/A-Z/;
    $str =~ tr/ａｂｃｄｅｆｇｈｉｊｋｌｍｎｏｐｑｒｓｔｕｖｗｘｙｚ/a-z/;
    $str
        =~ tr/\．\，\’\”\″\　\－\-\－\／\！\＃\％\＆\（\）\＊\＋\：\；\＜\＝\＞\？\
 ［\］\｛\｝/\.\,\' \-\-\-\/\!\#\%\&\(\)\*\+\:\;\<\=\>\?\[\]\{\}/;
    return $str;
}

#---------------------------------------------------------------------
# [rmSpaceSimple]  <安全> （1）先頭と文末 （2）読点まわり  だけ空白を削除
#---------------------------------------------------------------------
sub rmSpaceSimple {
    my $this = shift;
    my ($str) = @_;

    #== 先頭と文末のいらない空白を削除 ================
    $str =~ s/^\x{3000}+//;
    $str =~ s/\x{3000}+$//;
    $str =~ s/^\s+//;
    $str =~ s/\s+$//;

    #== 読点まわり ====================================
    while ( $str =~ /(、)(?:　+)/g ) {
        $str = "$`" . $1 . "$'";
    }
    while ( $str =~ /(?:　+)(、)/g ) {
        $str = "$`" . $1 . "$'";
    }
    return $str;
}

#---------------------------------------------------------------------
# [medcodHead]  MedCod テーブルの項目リストを引く
#---------------------------------------------------------------------
sub medcodHead{
    my $this = shift;
    my @medcod = ('CertificateKey',
      'LineNb',
      'TextLine',
      'CodeLine',
      'IntervalLine',
      'CodeOnly',
      'LineCoded');
    return \@medcod;
}

```

```
#---------------------------------------------------------------------
# [identHead]  Ident テーブルの項目リストを引く
#---------------------------------------------------------------------
sub identHead{
    my $this = shift;
    my @idents = ('CertificateKey',
      'LastChange',
      'DateBirth',
      'DateDeath',
      'Age',
      'Sex',
      'MannerOfDeath',
      'UCCode',
      'MainInjury',
      'Status',
      'Reject',
      'Coding',
      'CodingVersion',
      'CodingFlags',
      'SelectedCodes',
      'SubstitutedCodes',
      'ErnCodes',
      'AcmeCodes',
      'MultipleCodes',
      'Comments',
      'FreeText',
      'ToDoList',
      'CoderReject',
      'DiagnosisModified',
      'Residence',
      'Name',
      'Address',
      'AutopsyRequested',
      'AutopsyUsed',
      'RecentSurgery',
      'DateOfSurgery',
      'ReasonSurgery',
      'DateOfInjury',
      'PlaceOfOccurrence',
      'ActivityCode',
      'ExternalFreeText',
      'Pregnancy',
      'PregnancyContributeDeath',
      'Stillbirth',
      'MultiplePregnancy',
      'CompletedWeeks',
      'BirthWeight',
      'AgeOfMother',
      'ConditionsMother',
      'CertImage');
    return \@idents;
}

1;


```

Iris入力データ作成用 日本独自 ICDコード変換ルール

対象	変換後	数	対象症病名例	Valid Flag (2は確信度が低いもの)	対応方法・備考
J9699	J969	5338	呼吸不全	1	桁落とし
J9609	J960	4009	急性呼吸不全	1	桁落とし
E14	E149	1767	糖尿病	1	9合併症を伴わない
J9619	J961	1141	慢性呼吸不全	1	桁落とし
G903	G239	237	多系統萎縮症	2	シャイ・ドレーガー症候群 => G23.8, 多系統萎縮症 => G23.2 or G23.3
E11	E119	223	2型糖尿病	2	9合併症を伴わない
B59	J173	105	ニューモシスチス肺炎	1	併記されてる別コード
I7020	I702	31	末梢動脈硬化症	1	桁落とし
J9691	J969	30	2型呼吸不全	1	桁落とし
E10	E109	17	1型糖尿病	2	9合併症を伴わない
M7265	M726	10	外陰部壊死性筋膜炎	1	桁落とし
R17	R198	10	黄疸	1	消化器系および腹部に関するその他の明示された症状および兆候
J9611	J961	8	慢性2型呼吸不全	1	桁落とし
E13	E139	6	B型インスリン受容体異常症	2	9合併症を伴わない
J9601	J961	1	急性2型呼吸不全	1	桁落とし

Iris入力データ作成用期間表現正規化ルール

頻度	期間表現	Iris 表記	変数	単位	Valid Flag	memo
1253817	不詳			N/A	1	
533962	x日	xDays	$1	Days	1	
450503	約xヶ月	xMonths	$1	Months	1	
390626	約x年	xYears	$1	Years	1	
368978	xx日	xxDays	$1	Days	1	
364311	xヶ月	xMonths	$1	Months	1	
282086	x年	xYears	$1	Years	1	
271392	短時間		2	Hours	1	解釈
209669	不明			N/A	1	
207877	x日間	xDays	$1	Days	1	
206337	約x週間	xWeeks	$1	Weeks	1	
199044	約x日	xDays	$1	Days	1	
188073	x年xヶ月	x*12+xMonths	$1 * 12 + $2	Months	1	
162854	約x時間	xHours	$1	Hours	1	
161570	約xx年	xxYears	$1	Years	1	
143965	xx年	xxYears	$1	Years	1	
137907	xx日間	xxDays	$1	Days	1	
133864	x時間	xHours	$1	Hours	1	
115740	約x日間	xDays	$1	Days	1	
115673	約xx日	xxDays	$1	Days	1	
101846	xか月	xMonths	$1	Months	1	
100864	約x年xヶ月	x*12+xMonths	$1 * 12 + $2	Months	1	
99877	約x年間	xYears	$1	Years	1	
98383	x週間	xWeeks	$1	Weeks	1	
84620	約xか月	xMonths	$1	Months	1	
82609	約xケ月	xMonths	$1	Months	1	
75554	約xx日間	xxDays	$1	Days	1	
70673	約xカ月	xMonths	$1	Months	1	
67646	約xヵ月	xMonths	$1	Months	1	
64885	xケ月	xMonths	$1	Months	1	
61412	xカ月	xMonths	$1	Months	1	
60175	xヵ月	xMonths	$1	Months	1	
55943	x年xか月	x*12+xMonths	$1 * 12 + $2	Months	1	
53675	約xヶ月間	xMonths	$1	Months	1	
50109	x年間	xYears	$1	Years	1	
45212	約xx時間	xxHours	$1	Hours	1	
44825	xx時間	xxHours	$1	Hours	1	
38774	約xx年間	xxYears	$1	Years	1	
35001	数時間		2	Hours	1	解釈
31651	xx分	xxMinutes	$1	Minutes	1	
30922	x時間xx分	x*60+xxMinutes	$1 * 60 + $2	Minutes	1	
30801	x年xカ月	x*12+xMonths	$1 * 12 + $2	Months	1	
30338	x年xケ月	x*12+xMonths	$1 * 12 + $2	Months	1	
30091	x年xヵ月	x*12+xMonths	$1 * 12 + $2	Months	1	
28310	xxヶ月	xxMonths	$1	Months	1	
28113	約xx分	xxMinutes	$1	Minutes	1	
27672	約xxヶ月	xxMonths	$1	Months	1	
25236	xx年間	xxYears	$1	Years	1	
23092	x年xxヶ月	x*12+xMonths	$1 * 12 + $2	Months	1	

22188	約x年半	x*12+6Months	$1 * 12 + 6	Months	1	
20461	約x年xか月	x*12+xMonths	$1 * 12 + $2	Months	1	
17932	xヶ月間	xMonths	$1	Months	1	
16965	約x年xケ月	x*12+xMonths	$1 * 12 + $2	Months	1	
16438	数分		2	Minutes	1	解釈
16376	同上	前の期間と同じ		N/A	q	他セル(10,12,14,16:I*_p)情報利用
16360	数日		2	Days	1	解釈
16010	xヶ月xx日	31*x+xxDays	$1 * 31 + $2	Days	1	
15201	約x年xカ月	x*12+xMonths	$1 * 12 + $2	Months	1	
14242	数年		2	Years	1	解釈
14167	約x年xヵ月	x*12+xMonths	$1 * 12 + $2	Months	1	
12905	数日間		2	Days	1	解釈
12632	約x時間xx分	x*60+xxMinutes	$1 * 60 + $2	Minutes	1	
12573	約xヶ月半	31*x+15Days	$! * 31 + 15	Days	1	
11409	約半年	6Months	6	Months	1	
10246	約x年xxヶ月	x*12+xxMonths	$1 * 12 + $2	Months	1	
9957	約xケ月間	xMonths	$1	Months	1	
9909	数年間		2	Years	1	解釈
9737	短時間*推定+		2	Hours	1	解釈
9300	約x週	xWeeks	$1	Weeks	1	
8528	約xカ月間	xMonths	$1	Months	1	
8436	x月		$1	Months	1	解釈
8363	急死		1	Days	1	解釈
8343	xx年xヶ月	xx*12+xMonths	$1 * 12 + $2	Months	1	
8214	平成xx年x月x	逆算する			q	他セル（49:death_ymd）利用
8214	xxか月	xxMonths	$1	Months	1	
7598	約xか月間	xMonths	$1	Months	1	
7543	数ヶ月		3	Months	1	解釈
7406	x年半	x*12+6Months	$1 * 12 + 6	Months	1	
7375	約xヵ月間	xMonths	$1	Months	1	
7305	x年xxか月	x*12+xxMonths	$1 * 12 + $2	Months	1	
7106	約x年xヶ月間	x*12+xMonths	$1 * 12 + $2	Months	1	
6335	xヶ月半	x*31+15Days	$1 * 31 + 15	Days	1	
6256	数分間		2	Minutes	1	解釈
6213	即死		2	Minutes	1	解釈
6011	xxx日	xxxDays	$1	Days	1	
5999	約x時間半	x*60+30Minutes	$1 * 60 + 30	Minutes	1	
5740	xヶ月x日	31*x+xDays	$1 * 31 + $2	Days	1	
5721	長期間				q	不明と同義→短期間と比較
5563	約xxか月	xxMonths	$1	Months	1	
5373	約x月	xMonths	$1	Months	1	
5010	x週	xWeeks	$1	Weeks	1	
4992	約xxケ月	xxMonths	$1	Months	1	
4830	xxケ月	xxMonths	$1	Months	1	
4683	xxカ月	xxMonths	$1	Months	1	
4533	xか月xx日	x*31+xxDays	$1 * 31 + $2	Days	1	
4457	約半日	12Hours	12	Hours	1	
4425	約xxカ月	xxMonths	$1	Months	1	
4160	x年x月		$1 * 12 + $2	Months	1	解釈
4150	平成xx年xx月	逆算する			q	他セル（49:death_ymd）利用
4069	xxヵ月	xxMonths	$1	Months	1	
4041	xx時間xx分	xx*60+xxMinutes	$1 * 60 + $2	Minutes	1	

4037	x年xxカ月	x*12+xxMonths	$1 * 12 + $2	Months	1	
3971	数週間		3	Weeks	1	解釈
3875	約xxヵ月	xxMonths	$1	Months	1	
3837	記入なし			N/A	1	
3697	約x.xヶ月	x.x*31Days	$1	Months	1	解釈
3677	x年xxケ月	x*12+xxMonths	$1 * 12 + $2	Months	1	
3676	x年xxヵ月	x*12+xxMonths	$1 * 12 + $2	Months	1	
3671	約xx分間	xxMinutes	$1	Minutes	1	
3501	x年以上	xYears	$1	Years	1	
3475	x分	xMinutes	$1	Minutes	1	
3334	x年xヶ月間	x*12+xMonths	$1 * 12 + $2	Months	1	
3236	約xヵ月半	x*31+15Days	$1 * 31 + 15	Days	1	
3181	不詳-不詳			N/A	1	
3055	xx年以上	xxYears	$1	Years	1	
3028	xヵ月xx日	x*12+xxDays	$1 * 31 + $2	Days	1	
2987	xカ月xx日	x*12+xxDays	$1 * 31 + $2	Days	1	
2972	xケ月間	xMonths	$1	Months	1	
2912	約xx年xヶ月	xx*12+xMonths	$1 * 12 + $2	Months	1	
2878	直後			N/A	1	
2865	詳細不明			N/A	1	
2826	xか月間	xMonths	$1	Months	1	
2815	x.xヶ月	x.x*31Days	$1	Months	1	解釈
2772	xカ月間	xMonths	$1	Months	1	
2741	約xxヶ月間	xxMonths	$1	Months	1	
2719	xx年xか月	xx*12+xMonths	$1 * 12 + $2	Months	1	
2659	平成xx年x月	逆算する			q	他セル（49:death_ymd）利用
2615	空欄			N/A	1	
2590	xケ月xx日	x*31+xxDays	$1 * 31 + $2	Days	1	
2576	約半年間	6Months	6	Months	1	
2503	約xケ月半	x*31+15Days	$1 * 31 + 15	Days	1	
2434	xx分間	xxMinutes	$1	Minutes	1	
2433	約x分	xMinutes	$1	Minutes	1	
2430	xヵ月間	xMonths	$1	Months	1	
2409	まもなく			N/A	1	
2407	x時間半	x*60+30Minutes	$1 * 60 + 30	Minutes	1	
2407	約x年前	xYears	$1	Years	1	
2361	短期間		1	Months	1	解釈
2356	約xか月半	x*31+15Days	$1 * 31 + 15	Days	1	
2327	約xカ月半	x*31+15Days	$1 * 31 + 15	Days	1	
2325	数十分		30	Minutes	1	解釈
2199	約x.x年	x.x*12Months	$1	Years	1	解釈
2196	半年	6Months	6	Months	1	
2133	xヵ月半	x*31+15Days	$1 * 31 + 15	Days	1	
2118	約x年xxか月	x*12+xxMonths	$1 * 12 + $2	Months	1	
2081	約xx年前	xxYears	$1	Years	1	
1975	数時間*推定+		3	Hours	1	解釈
1929	約数時間		3	Hours	1	解釈
1903	x年前	xYears	$1	Years	1	
1832	約x年-約x年	xYears	$2	Years	1	解釈
1823	不祥			N/A	1	
1810	約xヶ月xx日	x*31+xxDays	$1 * 31 + $2	Days	1	
1772	約x年xxケ月	x*12+xxMonths	$1 * 12 + $2	Months	1	

1760	約x年xxカ月	x*12+xxMonths	$1 * 12 + $2	Months	1	
1679	約x.x時間	x.x*60Minutes	$1	Hours	1	解釈
1636	x時間x分	x*60+xMinutes	$1 * 60 + $2	Minutes	1	
1616	約x年x月	x*12+xMonths	$1 * 12 + $2	Months	1	
1584	xxxx年x月x	逆算する			q	他セル（49:death_ymd）利用
1581	xか月x日		$1 * 31 + $2	Days	1	
1563	数十年		20	Years	1	解釈
1562	まもなく*推定+					
1561	xか月半	x*31+15Days	$1 * 31 + 15	Days	1	
1559	x年xヶ月xx日	x*365+x*31	$1 * 12 + $2	Months	1	解釈
1535	xx年前	xxYears	$1	Years	1	
1522	即時					
1464	xxx日間	xxxDays	$1	Days	1	
1454	約x日半	x*24+12Hours	$1 * 24 + 12	Hours	1	
1433	xx年xカ月	xx*12+xMonths	$1 * 12 + $2	Months	1	
1425	約x年xxヵ月	xx*12+xMonths	$1 * 12 + $2	Months	1	
1395	xカ月半	x*31+15Days	$1 * 31 + 15	Days	1	
1393	半日	12Hours	12	Hours	1	
1392	xx年xヵ月	xx*12+xMonths	$1 * 12 + $2	Months	1	
1388	約x年xケ月間	x*12+xMonths	$1 * 12 + $2	Months	1	
1358	xx年xケ月	xx*12+xMonths	$1 * 12 + $2	Months	1	
1341	やや短時間					
1327	数か月	6Months	3	Months	1	解釈
1309	xケ月半	x*31+15Days	$1 * 31 + 15	Days	1	
1308	約xxx日	xxxDays	$1	Days	1	
1304	Hxx.x.xx	逆算する			q	他セル（49:death_ymd）利用
1272	平成xx年	逆算する			q	他セル（49:death_ymd）利用
1213	xヶ月xx日間	x*31+xxDays	$1 * 31 + $2	Days	1	
1208	数カ月		3	Months	1	解釈
1208	約数分		3	Minutes	1	解釈
1187	記載なし				q	不明のため
1187	短時間推定					
1167	約xx年-約x年	xYears	$1	Years	1	解釈
1164	瞬時					
1161	約x時間*推定+	xHours	$1	Hours	1	解釈
1161	x日前	xDays	$1	Days	1	
1156	xヵ月x日	x*31+xDays	$1 * 31 + $2	Days	1	
1146	xx年xxヶ月	xx*12+xxMonths	$1 * 12 + $2	Months	1	
1144	xxヶ月間	xxMonths	$1	Months	1	
1139	数ヶ月間	6Months	3	Months	1	
1131	ごく短時間					
1129	平成xx年x月xx日	逆算する			q	他セル（49:death_ymd）利用
1122	x日xx時間	x*24+xxHours	$1 * 24 + $2	Hours	1	
1112	x.x年	x.x*12Months	$1	Years	1	解釈
1070	xカ月x日	x*31+xDays	$1 * 31 + $2	Days	1	
1065	x時間程度	xHours	3	Hours	1	
1062	x年-x年	xYears	$2	Years	1	解釈
1007	x月xx日					
1001	約一週間	1Weeks	1	Weeks	1	
1001	x年xx日		$1 * 365 + $2	Days	1	
989	約xx年-約xx年	xxYears	$2	Years	1	解釈
974	x.x時間	x.x*60Minutes	$1	Hours	1	解釈

970	約x年xカ月間	x*12+xMonths	$1 * 12 + $2	Months	1	
965	xケ月x日	x*31+xDays	$1 * 31 + $2	Days	1	
953	xx月	xxMonths	$1	Months	1	
942	数ケ月		3	Months	1	
941	長期					
939	約x年xか月間	x*12+xMonths	$1 * 12 + $2	Months	1	
937	約xx月		$1	Months	1	
922	x～x時間		$1	Hours	1	
921	約x年xヵ月間		$1 * 12 + $2	Months	1	

機械学習による原死因コード変更有無予測結果（一部抜粋）

仮ID	予測結果 1:変更あり 0:変更なし	正解 1:変更あり 0:変更なし	仮原死因 (Iris) コード	Iris Reject	確定原死因 コード	外因符号	正誤
90	0	0	E46	No	E46		TRUE
560	0	0	I639	No	I639		TRUE
750	1	0	C349	No	C349		FALSE
1530	0	0	C679	No	C679		TRUE
2230	0	0	R54	No	R54		TRUE
2530	0	0	I619	No	I619		TRUE
2660	0	0	C169	No	C169		TRUE
3800	0	0	C23	No	C23		TRUE
4350	0	0	T58	No	T58	X474	TRUE
4760	0	0	I219	No	I219		TRUE
4830	0	0	D376	No	D376		TRUE
5370	0	0	R579	No	R579		TRUE
5730	0	0	J189	No	J189		TRUE
5810	0	0	R54	No	R54		TRUE
6070	0	0	R579	No	R579		TRUE
6110	0	0	C509	No	C509		TRUE
6180	0	0	C349	No	C349		TRUE
6190	0	0	R688	No	R688		TRUE
6870	0	0	C80	No	C80		TRUE
7620	0	0	C080	No	C080		TRUE
7840	1	1	I743	No	I709		TRUE
8280	0	0	F03	No	F03		TRUE
8630	0	0	C250	No	C250		TRUE
9410	0	0	C187	No	C187		TRUE
9890	0	0	C187	No	C187		TRUE
10300	0	1	I619	No	I611		FALSE
10380	1	0	S269	MayBe	S269	X788	FALSE
10490	0	0	I711	No	I711		TRUE
10650	1	1	C349	No	C343		TRUE
10700	0	0	F03	No	F03		TRUE
10730	0	0	I693	No	I693		TRUE
11030	0	0	C250	No	C250		TRUE
11320	0	0	T58	No	T58	X678	TRUE
11470	0	0	J189	No	J189		TRUE
12080	0	0	R54	No	R54		TRUE

12270	0	0	C169	No	C169		TRUE
12280	0	0	C55	No	C55		TRUE
12860	1	1	R090	No	T179	W790	TRUE
13320	0	0	C220	No	C220		TRUE
13520	0	0	I693	No	I693		TRUE
13550	0	0	N179	No	N179		TRUE
14830	0	0	I710	No	I710		TRUE
15010	0	0	R54	No	R54		TRUE
15510	0	0	J841	No	J841		TRUE
15570	0	0	R54	No	R54		TRUE
15590	0	0	C211	No	C211		TRUE
15730	0	0	C20	No	C20		TRUE
16030	0	0	C20	No	C20		TRUE
16200	1	0	C189	No	C189		FALSE
16900	0	0	C184	No	C184		TRUE
17490	0	0	C679	No	C679		TRUE
17540	0	0	T751	No	T751	W650	TRUE
17680	1	1	M069	No	I509		TRUE
18000	0	0	T751	No	T751	W650	TRUE
18160	1	1	I639	No	I693		TRUE
18970	0	0	R54	No	R54		TRUE
19340	0	0	C187	No	C187		TRUE
20390	0	0	C900	No	C900		TRUE
20550	0	0	C221	No	C221		TRUE
20710	0	0	C56	No	C56		TRUE
22340	0	0	J439	No	J439		TRUE
22700	0	0	M471	No	M471		TRUE
22740	0	0	I249	No	I249		TRUE
23290	0	0	I509	No	I509		TRUE
24520	0	0	C250	No	C250		TRUE
24670	0	0	C20	No	C20		TRUE
25100	0	0	C182	No	C182		TRUE
25150	0	0	I219	No	I219		TRUE
25340	0	1	C169	No	C162		FALSE
25580	0	0	I609	No	I609		TRUE
25630	0	0	I609	No	I609		TRUE
26710	0	1	C349	No	C343		FALSE
26750	0	0	C509	No	C509		TRUE
26770	0	0	J189	No	J189		TRUE
27100	0	0	I639	No	I639		TRUE
27180	0	0	Q042	No	Q042		TRUE

27280	0	0	I609	No	I609		TRUE
27520	0	0	I219	No	I219		TRUE
27590	0	0	N179	No	N179		TRUE
27760	0	0	C250	No	C250		TRUE
28510	0	0	C711	No	C711		TRUE
28570	0	0	I610	No	I610		TRUE
28660	0	0	C549	No	C549		TRUE
28780	0	1	I219	No	I251		FALSE
28910	0	0	I693	No	I693		TRUE
29470	0	0	C509	No	C509		TRUE
29540	0	1	T794	No	S409	W20	FALSE
29600	0	0	I609	No	I609		TRUE
29750	0	0	I619	No	I619		TRUE
29830	0	0	F03	No	F03		TRUE
29880	0	0	N189	No	N189		TRUE
30160	0	0	S065	No	S065	W010	TRUE
30670	0	0	C349	No	C349		TRUE
30710	0	0	C445	No	C445		TRUE
30780	0	0	I422	No	I422		TRUE
30810	0	0	C20	No	C20		TRUE
31480	0	0	K922	No	K922		TRUE
31710	0	0	J690	No	J690		TRUE
31800	0	0	R54	No	R54		TRUE
32520	1	0	C349	No	C349		FALSE
33560	0	0	S328	No	S328	W13	TRUE
33860	0	0	C159	No	C159		TRUE
34240	0	0	I509	No	I509		TRUE
34250	0	0	C169	No	C169		TRUE
34620	0	0	R54	No	R54		TRUE
35020	0	0	C539	No	C539		TRUE
36170	0	0	C184	No	C184		TRUE
36490	0	0	C159	No	C159		TRUE
36560	0	0	C159	No	C159		TRUE
36660	0	0	C349	No	C349		TRUE
36820	0	0	C162	No	C162		TRUE
36900	0	0	I211	No	I211		TRUE
37730	0	0	C221	No	C221		TRUE
37810	0	0	K746	No	K746		TRUE
38310	0	0	J690	No	J690		TRUE
38630	0	0	C20	No	C20		TRUE
38910	0	0	N19	No	N19		TRUE

38950	0	0	C220	No	C220		TRUE
39030	0	0	R54	No	R54		TRUE
39440	0	0	I219	No	I219		TRUE
39490	1	1	C248	No	C240		TRUE
40140	1	1	M625	No	R54		TRUE
41370	1	1	C248	No	C240		TRUE
41450	0	0	C184	No	C184		TRUE
41570	0	0	J189	No	J189		TRUE
41780	0	0	K802	No	K802		TRUE
41890	0	0	J690	No	J690		TRUE
41980	0	0	C240	No	C240		TRUE
42160	0	0	C20	No	C20		TRUE
42360	0	0	J690	No	J690		TRUE
42490	0	0	I693	No	I693		TRUE
42630	0	0	R54	No	R54		TRUE
43320	0	0	N390	No	N390		TRUE
43660	0	0	G301	No	G301		TRUE
44090	1	1	R090	No	T598	X670	TRUE
44120	0	0	J189	No	J189		TRUE
44380	0	0	I716	No	I716		TRUE
44450	0	1	R688	No	S828	X599	FALSE
44500	0	0	D381	No	D381		TRUE
44530	0	0	C509	No	C509		TRUE
45750	0	0	I619	No	I619		TRUE
46030	1	0	D65	No	D65		FALSE
46510	0	0	C169	No	C169		TRUE
46590	0	0	C187	No	C187		TRUE
46840	0	0	C65	No	C65		TRUE
46980	0	0	C220	No	C220		TRUE
47100	0	0	C20	No	C20		TRUE
47720	0	0	C509	No	C509		TRUE
47850	0	0	I219	No	I219		TRUE
48430	0	0	I48	No	I48		TRUE
48470	0	0	C169	No	C169		TRUE
48830	0	0	J449	No	J449		TRUE
48880	0	0	C349	No	C349		TRUE
48960	1	1	R571	No	T794	V031	TRUE
49030	1	1	J189	No	J690		TRUE
49050	0	0	C20	No	C20		TRUE
49080	0	0	I619	No	I619		TRUE
49310	0	0	E146	No	E146		TRUE

50020	0	0	I249	No	I249		TRUE
50620	1	0	E149	No	E149		FALSE
51040	0	0	R54	No	R54		TRUE
51210	0	0	I249	No	I249		TRUE
51260	0	0	J189	No	J189		TRUE
51460	0	0	J690	No	J690		TRUE
51670	1	1	C189	No	C180		TRUE
51870	0	1	J690	MayBe	I209		FALSE
52180	0	0	C20	No	C20		TRUE
52700	0	0	I249	No	I249		TRUE
52890	0	0	I509	No	I509		TRUE
52990	0	0	S062	No	S062	V031	TRUE
53380	0	0	C250	No	C250		TRUE
53460	0	0	T751	No	T751	V922	TRUE
53480	0	0	C169	No	C169		TRUE
53520	0	0	C220	No	C220		TRUE
53550	0	0	I509	No	I509		TRUE
53830	0	0	I259	No	I259		TRUE
53930	0	0	C187	No	C187		TRUE
55400	0	0	I714	No	I714		TRUE
55460	1	1	C349	No	C343		TRUE
55600	0	0	C169	No	C169		TRUE
55860	0	0	I609	No	I609		TRUE
55960	0	0	J189	No	J189		TRUE
56100	0	0	N19	No	N19		TRUE
57110	0	1	C259	No	C250		FALSE
57580	0	0	C349	No	C349		TRUE
57860	0	0	I712	No	I712		TRUE
58190	0	0	I219	No	I219		TRUE
58260	0	0	I219	No	I219		TRUE
58310	0	0	C169	No	C169		TRUE
58990	0	0	K562	No	K562		TRUE
59020	0	0	C187	No	C187		TRUE
59070	0	0	T751	No	T751	W65	TRUE
59260	1	0	C189	No	C189		FALSE
59530	0	0	D469	No	D469		TRUE
59810	0	0	I219	No	I219		TRUE
60090	0	0	C252	No	C252		TRUE
60110	0	0	R54	No	R54		TRUE
60190	0	0	I110	No	I110		TRUE
60250	0	1	C259	No	C220		FALSE

61570	0	0	J849	No	J849		TRUE
61640	0	0	C169	No	C169		TRUE
61990	0	0	J439	No	J439		TRUE
62250	0	0	C184	No	C184		TRUE
62290	0	0	K703	No	K703		TRUE
62520	1	1	C349	No	C348		TRUE
62640	0	0	C56	No	C56		TRUE
62890	0	0	K650	No	K650		TRUE
63140	0	0	C187	No	C187		TRUE
63700	0	0	C184	No	C184		TRUE
63850	0	0	C349	No	C349		TRUE
64300	0	0	J439	No	J439		TRUE
64930	0	0	C20	No	C20		TRUE
65080	1	1	C349	No	C343		TRUE
65300	0	0	C679	No	C679		TRUE
65400	0	0	K261	No	K261		TRUE
65690	0	0	I619	No	I619		TRUE
66060	0	0	C180	No	C180		TRUE
66230	1	0	C349	No	C349		FALSE
66670	0	0	C844	No	C844		TRUE
67190	0	0	C915	No	C915		TRUE
67220	0	0	S069	No	S069	W100	TRUE
67230	1	0	C349	No	C349		FALSE
67850	0	0	C169	No	C169		TRUE
67860	0	0	C341	No	C341		TRUE
68090	0	0	R54	No	R54		TRUE
68220	0	0	C509	No	C509		TRUE
68610	0	0	C220	No	C220		TRUE
68900	0	0	I619	No	I619		TRUE
69840	0	0	J459	No	J459		TRUE
70200	0	0	C509	No	C509		TRUE
70280	0	0	C349	No	C349		TRUE
71010	0	0	T07	No	T07	V234	TRUE
71080	0	0	C220	No	C220		TRUE
71110	0	1	M100	No	M103		FALSE
71790	1	1	C542	No	C55		TRUE
71880	0	0	C259	No	C259		TRUE
73990	1	1	J849	No	C341		TRUE
74040	0	0	S062	No	S062	W130	TRUE
74210	1	1	I620	No	S065	X599	TRUE
74490	0	0	R54	No	R54		TRUE

74630	0	0	I258	No	I258		TRUE
74830	0	0	I619	No	I619		TRUE
75630	0	0	C73	No	C73		TRUE
75670	0	0	J849	No	J849		TRUE
75790	0	0	C549	No	C549		TRUE
75810	0	0	C20	No	C20		TRUE
76080	1	1	I48	MayBe	N189		TRUE
76600	0	0	C060	No	C060		TRUE
77040	0	0	J849	No	J849		TRUE
77050	0	0	I609	No	I609		TRUE
77170	0	0	C859	No	C859		TRUE
77600	0	1		No	R99		FALSE
77660	0	0	C349	No	C349		TRUE
77760	0	0	F03	No	F03		TRUE
78020	0	0	C679	No	C679		TRUE
78230	0	0	J42	No	J42		TRUE
78380	0	0	R54	No	R54		TRUE
78620	0	0	M314	No	M314		TRUE
78850	0	0	I219	No	I219		TRUE
79140	0	0	C679	No	C679		TRUE
79160	0	0	C182	No	C182		TRUE
79240	1	1	I692	No	T179	W780	TRUE
79380	0	0	C20	No	C20		TRUE
79510	0	0	J189	No	J189		TRUE
79960	0	0	C169	No	C169		TRUE
80070	0	0	J189	No	J189		TRUE
80290	0	0	I713	No	I713		TRUE
81070	0	0	C169	No	C169		TRUE
81200	0	0	S062	No	S062	V031	TRUE
81570	0	1		No	R99		FALSE
82010	0	0	I249	No	I249		TRUE
82520	0	0	I710	No	I710		TRUE
82690	0	0	C20	No	C20		TRUE
83000	0	0	I340	No	I340		TRUE
83200	0	0	C159	No	C159		TRUE
83340	0	0	R54	No	R54		TRUE
83460	0	0	C169	No	C169		TRUE
83490	0	0	T751	No	T751	X71	TRUE
83620	0	1	S069	No	S064	X808	FALSE
84320	0	0	C240	No	C240		TRUE
84640	0	0	C159	No	C159		TRUE

85080	1	0	C349	No	C349		FALSE
85160	0	0	C64	No	C64		TRUE
85400	0	1	C169	No	C168		FALSE
85880	0	0	C182	No	C182		TRUE
86240	0	0	C187	No	C187		TRUE
86490	0	0	C169	No	C169		TRUE
86510	1	1	I714	No	I509		TRUE
86940	0	0	T751	No	T751	W650	TRUE
87440	0	0	I639	No	I639		TRUE
87470	0	0	I619	No	I619		TRUE
87600	0	0	R54	No	R54		TRUE
88010	0	0	C250	No	C250		TRUE
88090	1	1	C349	No	C341		TRUE
88110	0	0	E142	No	E142		TRUE
88180	0	0	C180	No	C180		TRUE
88190	0	0	I509	No	I509		TRUE
88310	0	0	I619	No	I619		TRUE
88480	1	1	C189	No	C187		TRUE
89010	1	0	C189	No	C189		FALSE
89190	0	0	C20	No	C20		TRUE
89510	0	0	T751	No	T751	W69	TRUE
90020	0	0	C509	No	C509		TRUE
90090	0	0	T751	No	T751	W650	TRUE
90350	0	0	I509	No	I509		TRUE
90670	0	0	I219	No	I219		TRUE
91180	0	0	C20	No	C20		TRUE
91220	1	1	S141	No	S197	X800	TRUE
91950	0	0	C679	No	C679		TRUE
92080	0	0	E669	No	E669		TRUE
92880	0	0	C180	No	C180		TRUE
93070	0	0	D619	No	D619		TRUE
93270	0	0	C509	No	C509		TRUE
93500	0	0	C159	No	C159		TRUE
93600	0	1	C20	No	C349		FALSE
93610	0	0	J42	No	J42		TRUE
93710	0	0	I509	No	I509		TRUE
94030	1	0	C189	No	C189		FALSE
94080	0	1	J841	No	J849		FALSE
94090	0	0	C719	No	C719		TRUE
94910	0	0	T751	No	T751	W650	TRUE
95890	0	0	T751	No	T751	W650	TRUE

95930	0	0	I219	No	I219		TRUE
96600	0	0	J189	No	J189		TRUE
97280	0	0	C169	No	C169		TRUE
97330	1	0	C189	No	C189		FALSE
97720	0	0	M318	No	M318		TRUE
97890	0	0	I219	No	I219		TRUE
98480	0	0	I500	No	I500		TRUE
98640	0	0	C169	No	C169		TRUE
98650	0	1	E149	No	E144		FALSE
98700	0	0	I219	No	I219		TRUE
98950	0	0	B181	No	B181		TRUE
99160	0	0	C169	No	C169		TRUE
99400	0	0	F03	No	F03		TRUE
99850	0	0	C182	No	C182		TRUE
100850	1	0	C349	No	C349		FALSE
101250	0	0	C169	No	C169		TRUE
101320	0	0	I490	No	I490		TRUE
101490	0	0	C20	No	C20		TRUE
101890	0	0	C679	No	C679		TRUE
102440	0	0	I509	No	I509		TRUE
102660	0	0	C56	No	C56		TRUE
102980	0	0	C240	No	C240		TRUE
102990	0	1	D150	No	C37		FALSE
103280	0	0	R54	No	R54		TRUE
103290	0	0	R54	No	R54		TRUE
104050	0	0	R54	No	R54		TRUE
104180	0	0	I219	No	I219		TRUE
104870	0	0	C20	No	C20		TRUE
105230	0	0	C162	No	C162		TRUE
105310	0	0	C240	No	C240		TRUE
105590	0	0	S299	No	S299	X80	TRUE
105790	0	1	C169	No	C160		FALSE
106570	0	0	I691	No	I691		TRUE
106830	0	0	I350	No	I350		TRUE
108180	1	1	K859	No	K851		TRUE
108640	0	0	I509	No	I509		TRUE
108970	0	0	D469	No	D469		TRUE
109160	0	0	I259	No	I259		TRUE
109240	1	1	C189	No	C187		TRUE
109790	0	0	S141	No	S141	V051	TRUE
110530	0	0	I619	No	I619		TRUE

110600	0	0	I609	No	I609		TRUE
110850	0	1	C169	No	C162		FALSE
110860	0	1	I48	No	I630		FALSE
110960	1	1	G309	No	J189		TRUE
111300	0	0	S099	No	S099	X599	TRUE
111820	0	0	I619	No	I619		TRUE
111850	0	0	R54	No	R54		TRUE
111960	0	0	I200	No	I200		TRUE
112090	0	0	I693	No	I693		TRUE
112170	0	0	J189	No	J189		TRUE
112290	0	0	C480	No	C480		TRUE
112460	0	0	R54	No	R54		TRUE
112580	0	0	T07	No	T07	W110	TRUE
112590	0	0	I639	No	I639		TRUE
112820	0	0	N390	No	N390		TRUE
113080	0	0	C221	No	C221		TRUE
114210	0	0	G20	No	G20		TRUE
114330	1	1	C189	No	C184		TRUE
114710	0	0	C220	No	C220		TRUE
114940	1	0	C19	No	C19		FALSE
115060	0	0	I48	No	I48		TRUE
115200	0	0	C240	No	C240		TRUE
115830	0	0	J189	No	J189		TRUE
116100	0	0	C679	No	C679		TRUE
116600	0	0	C169	No	C169		TRUE
117110	0	0	I509	No	I509		TRUE
117500	0	0	C65	No	C65		TRUE
118400	0	0	C220	No	C220		TRUE
118500	0	0	K746	No	K746		TRUE
118690	0	0	A162	No	A162		TRUE
118740	0	0	C23	No	C23		TRUE
118800	0	0	C549	No	C549		TRUE
119610	0	0	D329	No	D329		TRUE
119640	1	1	I259	No	I249		TRUE
119710	0	1	I619	No	I509		FALSE
119990	0	0	S099	No	S099	X808	TRUE
120020	0	0	G301	No	G301		TRUE
120230	0	0	K255	No	K255		TRUE
120280	0	0	J189	No	J189		TRUE
121010	0	0	C20	No	C20		TRUE
121450	0	0	T751	No	T751	W65	TRUE