

```

#!/usr/bin/perl -w
use strict;

my @ringB = ( "U", "V", "W", "X", "Y", "Z", "A", "B", "C", "D", "E", "F",
"G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T" );
my @ringC = ( "Z", "Y", "X", "W", "V", "U", "T", "S", "R", "Q", "P", "O",
"N", "M", "L", "K", "J", "I", "H", "G", "F", "E", "D", "C", "B", "A" );
my @key;
my $keyptr=0;
my @chars;

sub searchLetter {
    my $c = shift;
    my $i;

    for( $i=0; $i<26; $i++ ) {
        return $i if( $c eq $ringC[$i]);
    }

    return;
}

sub backRot {
    my @ta;
    my @sa;
    my $o = shift;
    my $i;

    @ta = (@ringC, @ringC);

    for( $i=0; $i<26; $i++ ) {
        push @sa, $ta[$i+$o];
    }

    @ringC = (@sa);
}

sub locateBC {
    my $c = shift;
    my $i;

    for( $i=0; $i<26; $i++ ) {
        if( $ringB[$i] eq $c ) {
            return $ringC[$i];
        }
    }

    return "*";
}

# Precond:
# ring C turned to starting point
# key without leading letter
sub enc {

```

```

my $offset;

print locateBC(shift @chars);

while( $#chars >=0 ) {
    backRot( $key[$keyptr++] );
    $keyptr=0 if( $keyptr> $#key );

    print locateBC(shift @chars);
}

}

$_ = <>;
chomp;
@key = split;

print "Key:";
for( my $i=0; $i<=#key; $i++ ) {
    print " ".$key[$i];
}
print "\n";

while(<>) {
    chomp;
    s/ //g;
    $_=~ tr/[a-z]/[A-Z]/;
    @chars = unpack "(A1)*";
}

# Frist step: sets initial state from letter

$key[0] = searchLetter( $key[0] );
backRot( $key[0] );
shift @key;

# Encrypt remainder with numbers

enc();

print "\n";

```