Box Puzzle

Version 1 – 01-21-2023

Pearl Initialization Header

```
#!/usr/bin/perl

use strict;
use warnings;
```

Initialize Variables

```
my @Box = (0..100);                          # Create a consecutive array with numbers from 0 to 100
                                             # Note: We won't use 0
my $random1 = 0;                             # Initialize random1 variable
my $random2 = 0;                             # Initialize random2 variable

my $Run_Attempts     = 1000;                 # Set some arbitrary number of attempts to a successful solution
my $Shuffle_Itterations = 100000;            # Set some arbitrary number to shuffle the deck
my $PathCount = 0;                           # Initialize PathCount variable
my $PathLimit = 50;                          # Initialize PathLimit variable
my $BoxIndex  = 1;                           # Initialize BoxIndex variable

my $Success = 0;
my $Attempts = 0;
my $PassAttempts = 0;
my $MaxPassAttempts = 0;
my $AvgPassAttempts = 0;
my $Samples = 0;
```

START of Main Loop

```
for ( $Attempts = 1; $Attempts <= $Run_Attempts; $Attempts++ ) {
```

Shuffle the Deck

```
        #---------------------------------------    # Shuffle the deck
        for ( my $i = 0; $i < $Shuffle_Itterations; $i++ ) {
                                                    # Generate two random numbers that are not equal
                $random1 = $random2 = 0;            # Initialize random1 and random2 to ZERO (or just the same number)
                while ( $random1 eq $random2){      # Continue to loop while random1 = random2; This insures an unequal end result
                        $random1 = int(rand(100)+1);# Generate first Random integer between 1 and 100
                        $random2 = int(rand(100)+1);# Generate second Random integer between 1 and 100
                }
        #---------------------------------------
                my $Inside_Box1 = $Box[$random1];   # Grab contents of Box1
                my $Inside_Box2 = $Box[$random2];   # Grab contents of Box2
                $Box[$random1] = $Inside_Box2;      # Swap contents of Box1 with Box2
                $Box[$random2] = $Inside_Box1;      # Swap contents of Box2 with Box1
        }
        #---------------------------------------
```

Show the Shuffled Boxes

```
        print "\n";
        print       "Box contents:\n";               # Show contents of Boxes
        for ( my $r = 1; $r < 11; $r++ ) {
                my $da = (($r-1)*10)+1;
                print "\$Box[".$da."]: ";
                for ( my $c = 1; $c < 11; $c++ ) {
                        if($c>1){ print ", "; }
                        print $Box[$c+(($r-1)*10)];
                }
                print "\n";
        }
```

Show the Loops

```
#----------------------------------------          # Show Loops

        $PassAttempts = 0;

        for ( $BoxIndex = 1; $BoxIndex <= 100; $BoxIndex++ ) {
                print "\n";
                $PathCount = 0;
                my $StartIndex = $BoxIndex;
                $BoxIndex = $Box[$BoxIndex];
                while($BoxIndex ne $StartIndex){
                        if($PathCount eq 0){
                                print "Loop #".$StartIndex." :";
                        }
                        print $BoxIndex." ";
                        $PathCount++;
                        last if $PathCount == 51;
                        $BoxIndex = $Box[$BoxIndex];
                }
                print "::".$PathCount;
                last if $PathCount == 51;
                $PassAttempts++;
        }
```

Error Checking

```
        if( $PathCount ne 51 ){
           $Success++ ;
        }
        if( $PathCount eq 51 ){
           if($PassAttempts gt $MaxPassAttempts){
                $MaxPassAttempts = $PassAttempts;
                $AvgPassAttempts = $AvgPassAttempts + $PassAttempts;
                $Samples++;
                }
        }
```

END of Main Loop

```
}
```

Show Results

```
print "\n\n";
my $p = (($Success*100)/$Run_Attempts);
print "$p% Success";
print "\n\n";
print "Max Pass Attempts before fail $MaxPassAttempts";
print "\n\n";
my $OutCome = $AvgPassAttempts / $Samples;
print "Average Pass Attempts before Fail $OutCome";
print "\n\n";
```