```
#!c:\users\...[your path]...\perl.exe

# Table of Sacred Cuts of the Square, by Vinyasi.
# Alias, searches for more silver ratios beyond what is commonly known about the Pell ratio.
# This file was originally formed on: 13 Dec. 2003 but whose principle
# was discovered earlier on a 1k RAM PC sometime between 1994 and 1997.
# No rights reserved. Copy left.
# Originally downloadable from: http://vinyasi.mayashastra.org/book/zip/sacred_square.zip
# Archived at: https://web.archive.org/web/20051029163755/ (this link is continued to next line...)
# http://vinyasi.mayashastra.org/book/zip/sacred_square.zip
# But currently downloadable from:
# http://vinyasi.info/Infinite%20Range%20of%20Golden%20Ratios/ (this link is continued to next line...)
# original%20research%20during%201994%20to%201997/

# use ">>" for appending to file or use ">" for erasing file and rewriting to it
open(OUT, ">", "data.txt") or die "open $data.txt failed: $!";

#print "Content-Type: text/html\n\n";

#<html><head><title>

print OUT "Sacred Square Cuts Among Even-Sided Polygons", "\n\n";

#</title>

use constant PI => 4 * atan2 1, 1;

$prynt = 1;
$shift = 100000;
# minimum search value for primes
$start = 2;
# maximum to search for primes which meet the criteria for a silver ratio; usually defaults to 5
$limit = 100;
# accuracy of search to this many digits
$safelimit = 10;

#<body bgcolor='black' text='white'><big>
#<center><h2>Sacred Square Cuts Among Even-Sided Polygons</h2></center>

print OUT "Search range is from ", $start, " to ", $limit, "\n\n";

for($x1 = $start; $x1 <= $limit; $x1++) { # even sides
  $prynt = 1;
  $mod = "false";
  for($x2 = $start; $x2 <= sqrt($x1); $x2++) {
    if($x1 % $x2 == 0) {
      $mod = "true";
      last; }}

  if($mod eq "false") {
    $read = "";
    $tot_sides = $x1 * 4; # even sides
    $readout = "\n". $x1. "p ". $tot_sides. "-Gon\n";
    $num_ang = $tot_sides / 2; # even sides
    $ang = 360 / $tot_sides; # even sides
    for($x3 = 1; $x3 <= $num_ang; $x3++) { # even sides
      $angle = $ang * $x3;
      $sin[$x3] = 2 * sin(PI * $angle / 360); # even sides
      $readout .= "Angle No.". $x3. ", Sin (". (int(int($angle * $shift + 1) / $shift));
      $readout .= " degrees / 2) = ". $sin[$x3]. " / 2\n"; }
      for($x4 = 1; $x4 <= $num_ang; $x4++) { # even sides
        for($x5 = ($x4 + 1); $x5 <= $num_ang; $x5++) { # even sides
          for($x7 = 1; $x7 <= $num_ang; $x7++) { # even sides
  $prynt = 1;
            for($x8 = ($x7 + 1); $x8 <= $num_ang; $x8++) { # even sides
              for($alt_sign = -1; $alt_sign <= 1; $alt_sign += 2) {
                $bsign = "- ";
                $csign = "+ ";
                $sin2sign = "+ ";
             for($alt1 = -1; $alt1 <= 1; $alt1 += 2) {
               for($alt2 = -1; $alt2 <= 1; $alt2 += 2) {
                 $alt3 = $alt1 * (-1);
                 $alt4 = $alt2 * (-1);
```

```
                 $al1 = '';
                 $al2 = '';
                 $al3 = '';
                 $al4 = '';
                 $al1 = "<sup>-</sup>" if($alt1 < 0);
                 $al2 = "<sup>-</sup>" if($alt2 < 0);
                 $al3 = "<sup>-</sup>" if($alt3 < 0);
                 $al4 = "<sup>-</sup>" if($alt4 < 0);
                 $sin1 = $sin[$x4] ** ($alt1) * $sin[$x5] ** ($alt3);
                 $sin2 = $sin[$x7] ** ($alt2) * $sin[$x8] ** ($alt4);
                 $be = $sin1 + ($alt_sign * $sin2);
                 $ce = $alt_sign * $sin1 * $sin2;
                 if($alt_sign < 0) {
                      $sin2 = abs($sin2);
                      $sin2sign = "- "; }
                    if($be < 0) {
                      $be = abs($be);
                      $bsign = "+ "; }
                    if($ce < 0) {
                      $ce = abs($ce);
                      $csign = "- "; }
                 if($be != 0 and length $be < $safelimit and length $ce < $safelimit and (length $sin1
>= $safelimit or length $sin2 >= $safelimit)) {
                   if(!$read) {
                    $read = "yes";
                    print OUT $readout, "\n"; }
                   $ae = 1;
                   if($ce =~ /\./) {
                    $ae *= 1 / $ce;
                    $be *= 1 / $ce;
                    $ce *= 1 / $ce; }
             if($prynt == 1) {
                   $prynt = 0;
                      print OUT "When the reciprocal of Angle No.", $x4, " (", 1 / $sin[$x4], ") is
multiplied by Angle No.", $x5, " (", $sin[$x5], "), then this equals the length of a diagonal: ";
                      print OUT $sin1, ".";
                      print OUT " Likewise, when Angle No.", $x7, " (", $sin[$x7], ") is multiplied by
the reciprocal of Angle No.", $x8, " (", 1 / $sin[$x8], "), then this yields the length of another
diagonal: ";
                      print OUT $sin2sign, $sin2, ".";
                      print OUT " And when the first diagonal is divided by the second diagonal, and when
the negation of the second diagonal is divided by the first diagonal, then this yields the two roots of a
quadratic polynomial: ";
                      print OUT "{", $sin1, ", ", $sin2sign, $sin2, "}", " = ";
                   print OUT $ae if($ae != 1);
                   print OUT "x^2 ", $bsign;
                   print OUT $be if($be ne '1');
                   print OUT "x ", $csign, $ce, ".\n\n";
             }}}}}}}}}
    undef @sin; }}

close OUT;

exit;
```