```
#!/opt/local/bin/perl

$infile = shift or die "No input file specified.";
$matlabFunctionName = shift or die "No Matlab function name specified.";

open INFILE ,"<$infile" or die "$!";

my @fields = ();
my $process = 0;
my $command = "";
my $x = [];
my $y = [];
my $row = 0;

while(my $cl = <INFILE>){
   $cl =~ s/(^\s+)|(\s+$)//g;
   @fields = split /\s+/, $cl;

   $command = $fields[-1];

   if($command eq 'newpath'){
      $process = 1;
      next;
   }
   if($command eq 'grestore'){
      last;
   }

   if($process){
      if($command eq 'moveto'){
         $x->[$row]->[0] = $fields[0];
         $y->[$row]->[0] = $fields[1];
      }
      elsif($command eq 'lineto'){
         $x->[$row]->[1] = $x->[$row]->[0];
         $y->[$row]->[1] = $y->[$row]->[0];

         $x->[$row]->[2] = $fields[0];
         $x->[$row]->[3] = $fields[0];

         $y->[$row]->[2] = $fields[1];
         $y->[$row]->[3] = $fields[1];

         $row++;

         #initialize the next row
         $x->[$row]->[0] = $x->[$row - 1]->[3];
         $y->[$row]->[0] = $y->[$row - 1]->[3];
      }
      elsif($command eq 'curveto'){
         $x->[$row]->[1] = $fields[0];
         $y->[$row]->[1] = $fields[1];

         $x->[$row]->[2] = $fields[2];
         $y->[$row]->[2] = $fields[3];

         $x->[$row]->[3] = $fields[4];
         $y->[$row]->[3] = $fields[5];

         $row++;

         #initialize the next row
         $x->[$row]->[0] = $x->[$row - 1]->[3];
         $y->[$row]->[0] = $y->[$row - 1]->[3];
      }
   }
}

#Both arrays will have one extra row due to steps wich initialize
#the "next" row. Note, this assumes that processing doesn't end on a
#moveto.
pop @$x;
pop @$y;
```

```
print "function [x, y] = $matlabFunctionName ()\n";
print "x = [\n";
for(my $i=0; $i < scalar(@$x); $i++){
   print "          ";
   for(my $j=0; $j < 4; $j++){
      print $x->[$i]->[$j], " ";
   }
   print ";\n";
}
print "];\n";

print "y = [\n";
for(my $i=0; $i < scalar(@$x); $i++){
   print "          ";
   for(my $j=0; $j < 4; $j++){
      print $y->[$i]->[$j], " ";
   }
   print ";\n";
}
print "];\n";
print "end\n";
```