```
#!/usr/bin/perl
print "Content-type: text/html\n\n";

# Get Query Variables see parseform.pl for comments
   $buffer = $ENV{"QUERY_STRING"};
    @pairs = split(/&/, $buffer);
     foreach $pair (@pairs) {
      ($nume, $value) = split(/=/, $pair);
        $value =~ tr/+/ /;
        $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
        $value =~ s/<!--(.|\n)*-->//g;
        $query{$nume} = $value;
   }
$repeat = 0;
 $value = $query{topgraph};
 while ($repeat <= 1){
                if ($value eq 'nitrogen'){
                        $title = "Nitrogen_Concentration_mg/L";
        }
        elsif ($value eq 'phosphorus'){
                        $title = "Phosphorus_mg/L";
        }
        elsif ($value eq 'sediments'){
                        $title = "Sediments_mg/L";
        }
        elsif ($value eq 'heavymetal'){
                        $title = "Heavy_Metal_mg/L";
        }
        elsif ($value eq 'toxins'){
                        $title = "Toxins_mg/L";
        }
        elsif ($value eq 'watertemp'){
                        $title = "Water_Temperature_Degrees_Celsius";
        }
        elsif ($value eq 'temperature'){
                        $title = "Air_Temperature_Degrees_Celsius";
        }
        elsif ($value eq 'precip'){
                        $title = "Precipitation_Inches";
        }
        elsif ($value eq 'ph'){
                        $title = "pH";
        }
        elsif ($value eq 'do'){
                        $title = "Dissolved_Oxygen_mg/L";
        }
        elsif ($value eq 'bod'){
                        $title = "BOD_mg/L";
        }
        elsif ($value eq 'saturateddo'){
                        $title = "Saturated_DO_mg/L";
        }
        elsif ($value eq 'runoff'){
                        $title = "Runoff_Meters_Cubed";
        }
        elsif ($value eq 'totalflow'){
                        $title = "Total_Flow_Meters_Cubed";
        }
        elsif ($value eq 'groundwater'){
                        $title = "Groundwater_Meters_Cubed";
        }
        else {
                        $title = "No_Title_Available";
        }
        if ($repeat == 0){
                $topgraph = $title;
        }
        if ($repeat == 1){
                $bottomoption = $title;
        }
        $value = $query{bottomoption};
        $repeat += 1;
}

#foreach $dog(sort keys %query){
#print "$dog: $query{$dog}<br>\n";
#}

$pid = $query{pid};
$process = $pid;
$pid1 = $query{pid};
$pid2 = ($query{pid} + 1);
$start = $query{topfrom};
$end = $query{topto};
$region = "Top vs Bottom selections";
# print "ID's: $pid, $pid1, $pid2";
$data1 = "/usr/local/jail/www/htdocs/tmpgif/riverout" . $pid1 . ".dat";
$data2 = "/usr/local/jail/www/htdocs/tmpgif/riverbout" . $pid1 . ".dat";
$data3 = "/usr/local/jail/www/htdocs/tmpgif/riverout" . $pid2 . ".dat";
$data4 = "/usr/local/jail/www/htdocs/tmpgif/riverbout" . $pid2 . ".dat";

# open the first two datafiles and read into new array
open (DATA1, "<$data1") || die " couldn't open file $data1: $!";
        while (<DATA1>){
                chop;
                if ($_ !~ /#/){

                        ($buffer, $index, $value) = split(/\s+/, $_);

                        # it's either going to be buffer or index
                        # or index and value.

                        if ($index){
                                #print "Adding Line: $. file 1";
                                push @data1, $value;
                        }
                        else {
                                #print "didn't push \"$value\" in data1";
                                ;
                        }
                }
        }
close (DATA1);
open (DATA2, "<$data3") || die " couldn't open file $data3: $!";
        while (<DATA2>){

                chop;
                if ($_ !~ /#/){

                        ($buffer, $index, $value) = split(/\s+/, $_);

                        if ($index){
                                #print "Adding Line: $. file 2";
                                push @data2, $value;

                        }
                        else {
                                #print "didn't push \"$value\" in data2";
                                ;
                        }
                }
        }
```

```
close (DATA2);

if ($query{changed} eq "1"){

open (DATA3, "<$data2") || print "couldn't open file $data2: $!";
        while (<DATA3>){
            chop;
                        if ($_ !~ /#/){

                                ($buffer, $index, $value) = split(/\s+/, $_);

                                # may have to remove this because of blank    sp
aces
                                if ($index){
                                        #print "$value\n";
                        push @data3, $value;

                                }
                                else {
                                        #print "didn't push \"$value\" in data3"
;
                                        ;
                                }
                        }
        }
close (DATA3);
open (DATA4, "<$data4") || print "couldn't open file $data4: $!";
        while (<DATA4>){

            chop;
                        if ($_ !~ /#/){
                                ($buffer, $index, $value) = split(/\s+/, $_);

                                if ($index){
                                        #print "$value\n";
                        push @data4, $value;
                                }
                                else {
                                        #print "didn't push \"$value\" in data3"
;
                                        ;
                                }
                        }
        }
close (DATA4);


}

$scatter1 = "/tmp/scatter1" . $pid . ".dat";
open (SCATTER1, ">$scatter1") || print "couldn't open file $scatter1: $!";
while ($i <= $#data1){
#       print "Data: $i $data1[$i] $data2[$i]\n<br>";
        if (($data2[$i]) && ($data1[$i])){
                print SCATTER1 "$data2[$i] $data1[$i]\n";
        }
        #print "SCATTER1 \"$data2[$i] $data1[$i]\n\"\n";
        $i++;
}
close (SCATTER1);
$i = 0;
$scatter2 = "/tmp/scatter2" . $pid . ".dat";
open (SCATTER2, ">$scatter2") || print "couldn't open file $scatter2: $!";

while ($i <= $#data3){
      # print "Data: $i $data3[$i] $data4[$i]\n<br>";
        if (($data4[$i]) && ($data3[$i])){
                print SCATTER2 "$data4[$i] $data3[$i]\n";
        }
        #print "SCATTER2 \"$data4[$i] $data3[$i]\n\";";
        $i++;
}
close (SCATTER2);

### DEBUG: print a line of a file

#       $i = 0;
#       open (SCATTER1, "<$scatter1") || print "couldn't open file $scatter2: $!
";
#       while (<SCATTER1>){

#               $i++;

#               print $_ if $i eq 5;

#       }
#       close (SCATTER1);

#       $i = 0;

###


# print datafile to tmp directory

# if 2nd datafiles exist,
#       open the second two datafiles and read into new array
#       print 2nd datafile to tmp directory

# call gnuplot
# remove files in tmp

$gnuplot = "/usr/local/bin/gnuplot";
$ppmtogif = "/usr/local/bin/ppmtogif";

        open (WRITEME, "| $gnuplot");
        print WRITEME "set terminal pbm small color\n";
        print WRITEME "set output \"/tmp/riverscatter$process.pbm\"\n";
system ("chmod 0666 /tmp/riverscatter$process.pbm");
        print WRITEME "set size .8,.8\n";
        print WRITEME "set autoscale\n";
#       print WRITEME "test\n";
#       print WRITEME "set nokey\n";
        print WRITEME "set key below\n";
        print WRITEME "set ylabel \"$topgraph\"\n";
        print WRITEME "set xlabel \"$bottomoption\"\n";
        print WRITEME "set timestamp top rotate\n";
#       print WRITEME "set xrange [$start:$end] \n";
#       print WRITEME "set yrange [0:10] \n";
        print WRITEME "set title \"$region\"\n";

if ($query{changed} eq "1"){

        print WRITEME

"plot \"$scatter1\" title \"Original\" w points lt 7 ps .6, ";

        print WRITEME
"\"$scatter2\" title \"Best_Management\" w points lt 9 ps .6\n";
}
else {
       print WRITEME
```

```
"plot \"$scatter1\" title \"Original\" w points lt 7 ps .6\n";

}
#       print WRITEME

#"\"$data3\" title \"Original_B\" w linespoints lt 2 ps .5,";
#       print WRITEME

#"\"$data4\" title \"Best_Management_B\" w l lt 6\n";

        print WRITEME "exit\n";
        close (WRITEME);

 system ("$ppmtogif < /tmp/riverscatter$process.pbm > /tmp/riverscatter$pid.gif"
);
 system ("mv /tmp/riverscatter$pid.gif /usr/local/jail/www/htdocs/tmpgif/");
 system ("rm $scatter2");
 system ("rm /tmp/riverscatter$pid.pbm");
 system ("rm $scatter1");
$gif = ("riverscatter" . $pid . ".gif");




print <<"HTML";

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
        <title>$topgraph vs $bottomoption</title>
</head>
<body bgcolor="#ffffff">
<img src="http://mvhs.mbhs.edu/tmpgif/riverscatter$pid.gif">


</body>
</html>

HTML

#foreach $dog(sort keys %query){
#print "$dog: $query{$dog}<br>\n";
#}



# print "Location: $graphic\n\n";
```