**Lab Assignment 2**

Write a PERL script for the following:

1. LCM and HCF

Answer:

print "Enter two numbers";

$x=<STDIN>;

$y=<STDIN>;

$a=$x;

$b=$y;

while($b!=0){

$t=$b;

$b=$a%$b;

$a=$t;

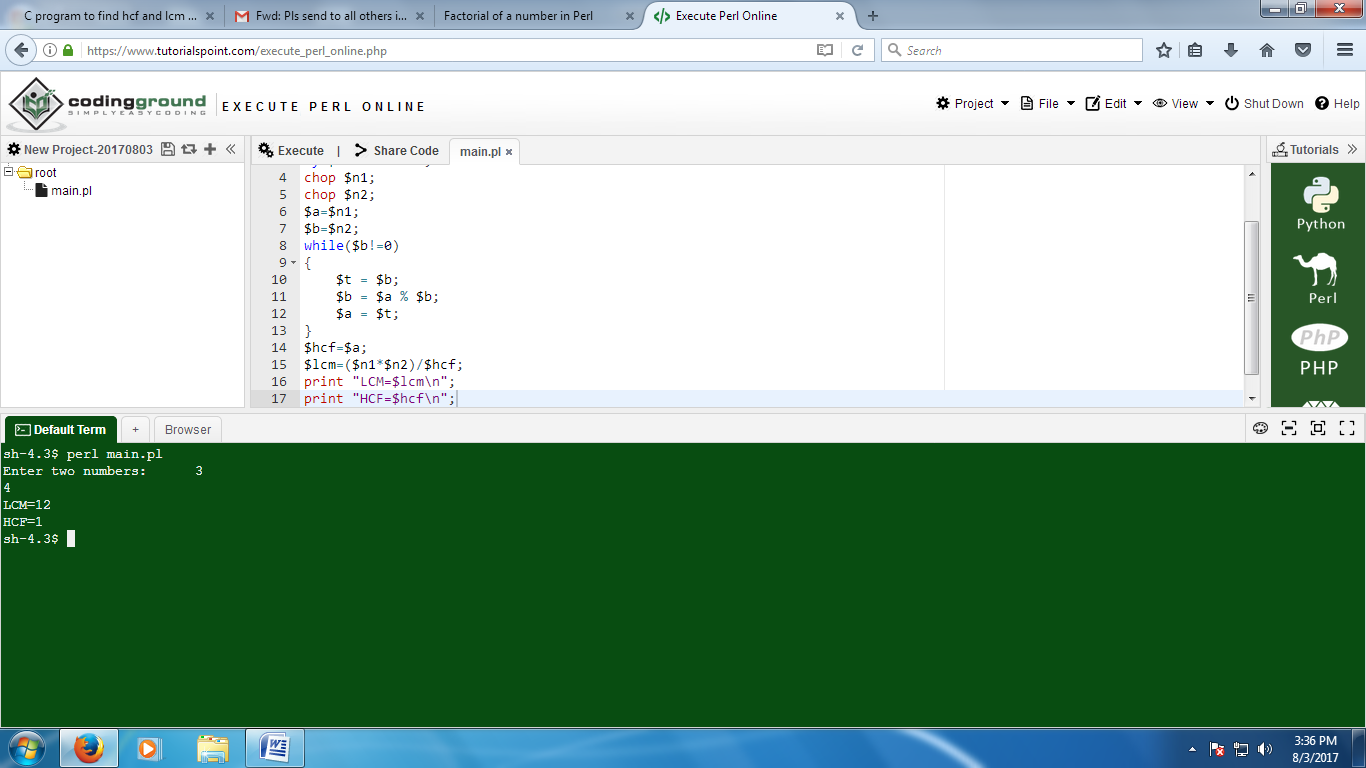
}

$gcd=$a;

$lcm=($x\*$y)/$gcd;

print "\nGreatest common divisor ".$gcd;

print "\n\nLeast common multiple ".$lcm;



2. Calculate area and perimeter of a triangle

Answer:

print "Enter three sides of triangle";

$a=<STDIN>;

$b=<STDIN>;

$c=<STDIN>;

$perimeter = $a+$b+$c;

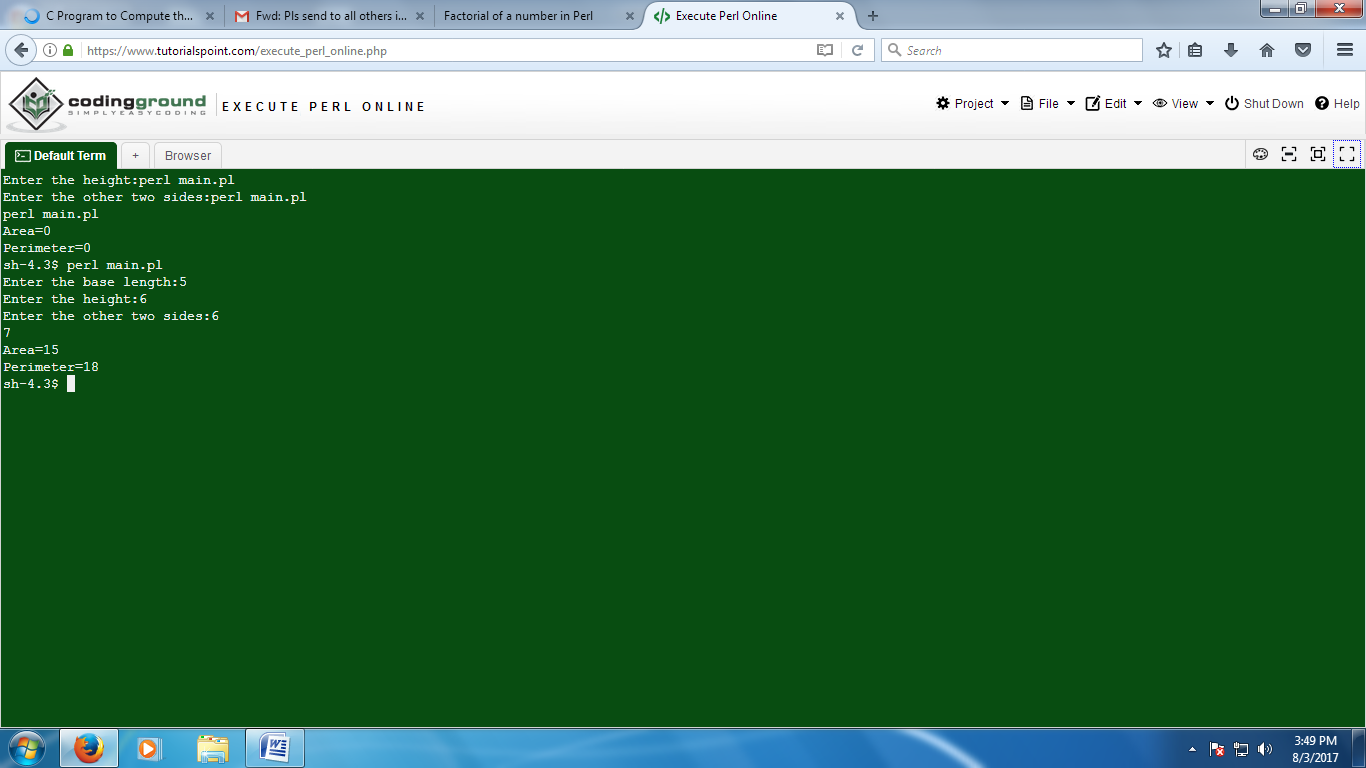
$s = ($a+$b+$c)/2;

$t=$s\*($s-$a)\*($s-$b)\*($s-$c);

$area=sqrt($t);

print "\n Perimeter of Triangle ".$perimeter;

print "\n Area of triangle ".$area;



3. Find sum of all digits in a number

Answer:

print “Enter your number:”;

my $num=<STDIN>;

chop $num;

$temp=$num;

while($num=0){

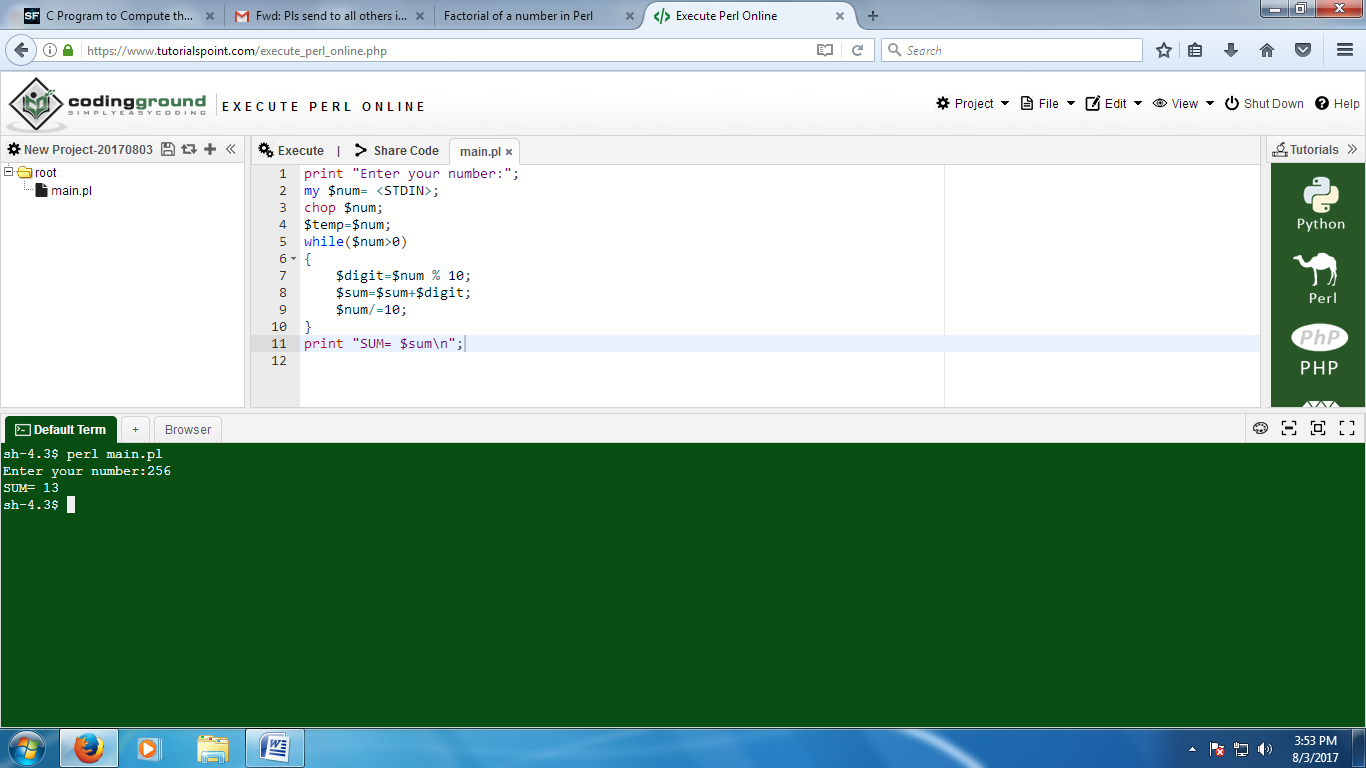
$digit=Snum % 10;

$sum=$sum+$digit;

$num/=10;

}

print “The sum is:”.$sum;



4. Create an array with 10 elements

1. Print the highest index of the array
2. Assign beyond the end of the array, to an element at index 20
3. Save the current highest index in a scalar, and print it.
4. Set the array size to 5 elements (index 4)
5. Print the array
6. Set the array size back to the previous size (using the scalar created in (d)).
7. Print the array

Answer:

#!/usr/bin/env

perl@arr=();

print "enter the input\n";

for($i=0;$i<10;$i++){

$n=<STDIN>;

chop($n);

$arr[$i]=$n;

}

print "Highest Index is ".$#arr."\n";

print "enter the input for index 20\n";

$n=<STDIN>;

chop($n);

@arr[20]=$n;

print "\nValue at Index 20 is ".$arr[20]."\n";

$count=$#arr;

print "high is ".$#arr."\n";

print "Maximum index is: ",scalar @arr,"\n";

$#arr=4;

print @arr;

$#arr=$count;

print @arr;

5. Create a program that prints a text value a variable number of times. Prompt the user to enter both the text and the number of times it is to be printed.

Answer:

#!/usr/bin/env perl

print "Enter the number:\n";

$num=<STDIN>;

chop $num;

print "Enter the variable:\n";

$str=<STDIN>

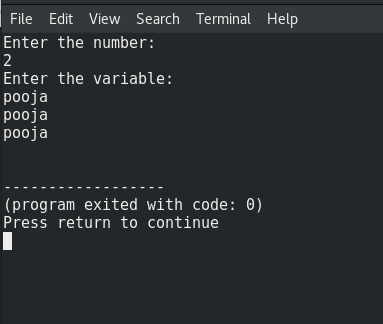
chop $str;

for(my $i=0;$i<$num;$i++){

print $str;

print "\n";

}



6. Write a subroutine to find maximum number in an array.

Answer:

#!/usr/bin/env

perl@arr=(2,7,4,1,9,0);

maxi (@arr);

sub maxi{

$last=$arr[0];

for $i(@arr){

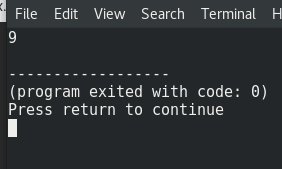
if($arr[$i]>$last){

$last=$arr[$i];

}

}

print $last;}



7. Write a subroutine to find positive and negative numbers and sum of positive and negative numbers in an array.

Answer:

#!/usr/bin/env

perl@arr=(1,7,-4,-6,7,-8);

print "The given array is:\n";

print @arr;

sum\_arr(@arr);

sub sum\_arr{

$pos=0;

$neg=0;

for ($i=0;$i<6;$i++){

if($arr[$i]>0){

$pos=$pos+$arr[$i];

}

Else

{

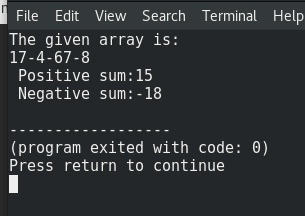
$neg=$neg+$arr[$i];

}

}

print "\n Positive sum:",$pos;

print "\n Negative sum:",$neg;

}

8.  Create a queue and perform various operations on it.

Answer:

print "Making a Queue\n";

@queue = qw( hi hello welcome );

print "Initial queue:\n @queue \n";

unshift(@queue, "ola");

print "Add item to queue:\n @queue \n";

$item = "hey";

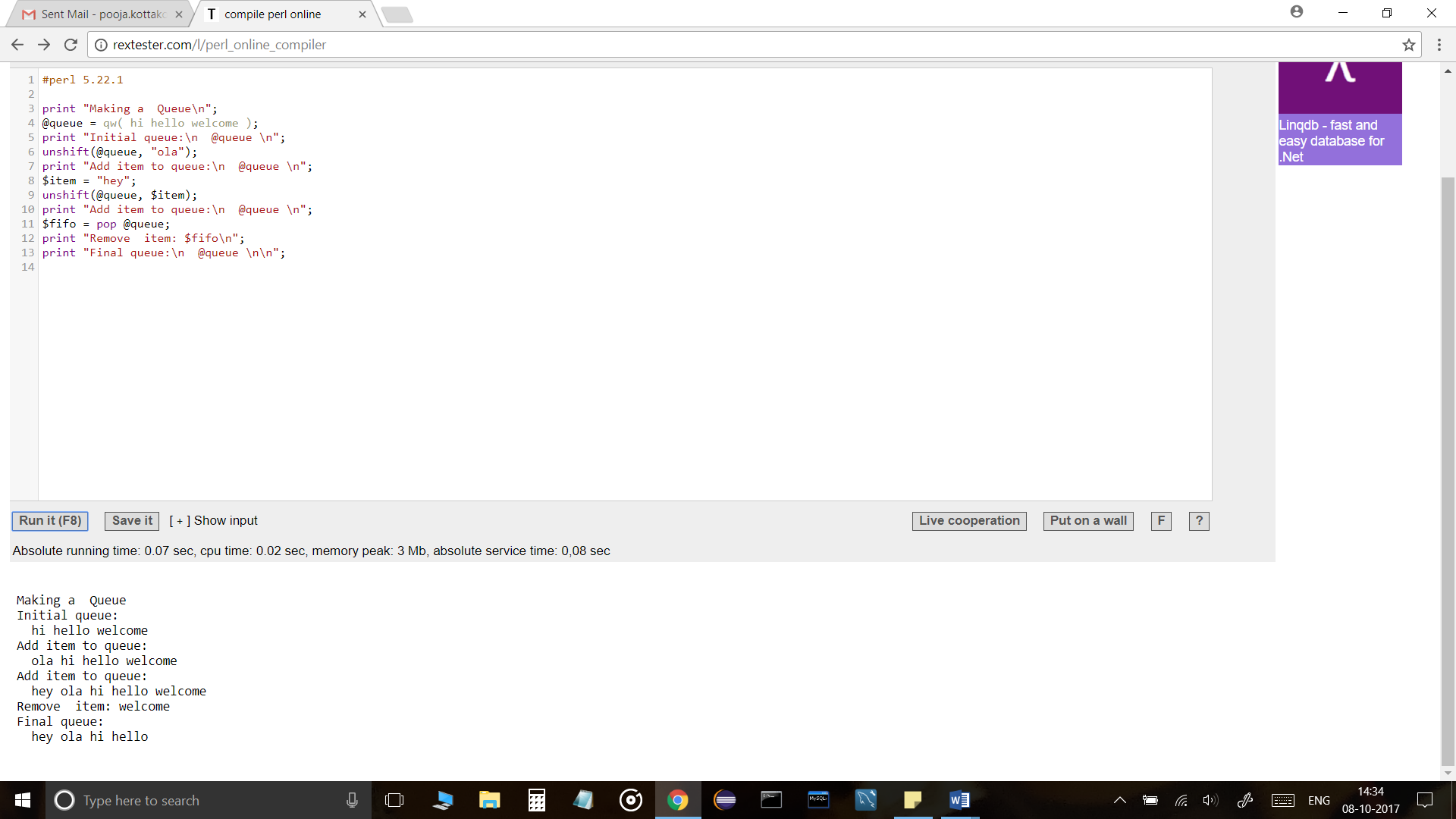
unshift(@queue, $item);

print "Add item to queue:\n @queue \n";

$fifo = pop @queue;

print "Remove item: $fifo\n";

print "Final queue:\n @queue \n\n";



9.   Create a new file a given name in a given directory.

1.Delete a file from given directory.

2.Move a file from given source location to a given destination location.

3.Copy contents of a source file to destination file.

4. Update existing file with given contents. (Append contents to the file).

5. Update permissions of a file.

6. Display all files from a given directory.

7. Display files permissions of all files from a given directory.

8. Display the contents of a given file.

Answer:

my $existingdir = 'C:/my';

mkdir $existingdir unless -d $existingdir;

open my $fileHandle, ">>", "$existingdir/myfile.txt" or die "Can't open!!”;

print $fileHandle "Welcome to Perl!\n";

close $fileHandle;

1. my $file = "myfile.txt";

unlink $file;

if(-e $file) {

print "File still exists!";

}

else

{

print "File deleted!";

}

2. $s = "C:/my/$myfile";

$d = "C:/not/"$notmyfile”;

`move $s $d`;

-e "$d/$myfile" && !-e $s ? print "Moved successfully\n" : print "Error $d/$myfile \n".$!;

3. #!/usr/bin/perl

my $infile1='myfile.txt';

my $infile2= ;

my $outfile= 'output.txt';

open(INFO,"<$infile1") or die "could not open '$infile1;

open(FILEA,"outputfromvariantparser1.txt") or die "could not open file";

open(FILEB,"<<output.txt") or die "could not open file";

my @raw\_data=<Filea>;

while(my $line=<INFO>){

chomp($line);

foreach $combination(@raw\_dat){

$line=~s/TEXT TO BE REPLACED/$combination/s;

}

print Fileb $line;

}

4. my $filename = 'myfile.txt';

open(my $fh, '>>', $filename) or die "Could not open file '$filename' $!";

say $fh "My first report generated by perl";

close $fh;

5. my $filename=”myfile.txt”;

chmod 0755, $filename;

6. opendir my $dir, "/my" or die "Cannot open directory: $!";

my @files = readdir $dir;

closedir $dir;

7. $permissions = S\_IMODE($mode);

8. my $filename = 'myfile.txt';

open(my $fh, $filename)

or die "Could not open file '$filename' $!";

while (my $row = <$fh>) {

chomp $row;

print "$row\n";

}

10.Using the subroutines, write a program that reads a text ﬁle and counts the number of times each word is used.

Answer:

my %count;

my $file = shift or die "Usage: $0 FILE\n";

open my $fh, '<', $file or die "Could not open '$file' $!";

while (my $line = <$fh>) {

chomp $line;

foreach my $str (split /\s+/, $line) {

$count{$str}++;

}

}

foreach my $word (reverse sort { $count{$a} <=> $count{$b} } keys %count) {

printf "%-31s %s\n", $word, $count{$word};

}