1. Write a PERL program to convert a decimal number (supplied as argument) to an unsigned binary number (for example if the argument is 22, then the output should 23 **be 10110). If an argument is present, it can be a valid decimal number and if no argument is present, the program should display an error message.**

```
#!/usr/bin/perl
$pow=1;
$dec=0;
die("no arguments\n")if(@ARGV==0);
foreach $num(@ARGV)
{
$bin=$num;
until($num==0)
{
$bit=$num%10;
$dec=$dec+($bit*$pow);
$pow=$pow*2;
$num=$num/10;
}
print("the decimal no. of $bin is $dec\n");
$pow=1;
$dec=0;
}
```

Output:

```
perl ty11.pl
no arguments
perl ty11.pl 11010
the decimal no. of 11010 is 26
```

2. PERL script that echoes its command line arguments, one per line after translating all lower case letters to upper case.

```
#! /usr/bin/perl
die("you have not entered the arguments\n")if(@ARGV==0);
foreach $arg(@ARGV)
{
$arg=~ tr/a-z/A-Z/;
printf("$arg\n");
}
```

Output:

```
perl a.pl linux internals lab
LINUX
INTERNALS
LAB
```

3. PERL program to find the sum of digits of an unsigned number passed through argument.

```
#!/usr/bin/perl
foreach $num (@ARGV)
{
$original_no=$num;
```

```
until($num==0)
{
$digit=$num%10;
$sum=$sum+$digit;
$num=int($num/10);
}
print(“sum of digits of $original_no is $sum”);
```

Output:

```
perl sumofdigits.pl
12345
sum of digits of 12345 is 15
```

4. Write a PERL program that prompts user to input the string and a number, and prints the string that many times, with each string on separate line..

```
#!/usr/bin/perl
print “string:”
$a=<STDIN>;
print “number of times string to be displayed:”
chop($b=<STDIN>);
$c=$ax$b;
print “result is:\n$c”;
```

Output:

```
perl input.pl
string: Nitte
number of times string to be displayed:2
result is: Nitte
Nitte
```