1)Create Hello World program

```
#! /usr/bin/perl -w
print "Hello World \n ";
```

```
,---(12:33:38)--[/root]
+-root@localhost> ./hello.pl
Hello World
```

2)Print the perl script arguments

```
#! /usr/bin/perl -w
$num=$#AVGV+1;
print("no of total arguments is $num\n");
```

```
+-root@localhost> ./hello.pl
Name "main::AVGV" used only once: possible typo at ./hello.pl line 2.
no of total arguments is 0
```

3)Write a program that computes the circumference of a circle with a radius of 12.5.

```
#! /usr/bin/perl -w
$radius=12.5;
$pi=3.14;
$circumference=2*$pi*$radius;
print"the circumference of a circle with a radius of 12.5 =$circumference \n";
```

```
,---(12:39:55)--[/root]
+-root@localhost> ./hello.pl
the circumference of a circle with a radius of 12.5 =78.5
```

4)Modify the program from the previous exercise to prompt for and accept a radius from the person running the program.

```
#! /usr/bin/perl -w
print("enter the radius \n");
chomp($radius=<STDIN>);
$pi=3.14;
$circumference=2*$pi*$radius;
print"the circumference of a circle with a radius of $radius =$circumference \n";
```

```
,---(12:43:30)--[/root]
+-root@localhost> ./hello.pl
enter the radius
10
the circumference of a circle with a radius of 10 =62.8
```

5)Write a program that prompts for and reads two numbers, and prints out the result of the two numbers multiplied together.

```
#! /usr/bin/perl -w
print("enter the first number \n");
chomp($num1=<STDIN>);
print("enter the second number \n");
chomp($num2=<STDIN>);
$result=$num1*$num2;
print"the result of $num1 * $num2 =$result \n";
```

```
,---(12:43:50)--[/root]
+-root@localhost> ./hello.pl
enter the first number
5
enter the second number
3
the result of 5 * 3 =15
```

6)Write a program that reads a string and a number, and prints the string the number of times indicated by the number on separate lines.

```
#! /usr/bin/perl -w
print("enter the number \n");
chomp($num=<STDIN>);
print("enter the string \n");
chomp($str=<STDIN>);
if($num<0){
print("enter positive number \n");
chomp($num=<STDIN>);}
while($num>0){
print("the string is $str\n");
$num--;
}
```

```
,---(13:03:21)--[/root]
+-root@localhost> ./hello.pl
enter the number
3
enter the string
Omnia
the string is Omnia
the string is Omnia
the string is Omnia

,---(13:04:45)--[/root]
+-root@localhost> ./hello.pl
enter the number
-4
enter the string
ppp
enter positive number
2
the string is ppp
the string is ppp
```

7)Write a program that reads a list of strings on separate lines and prints out the list in reverse order.

```
#! /usr/bin/perl -w
print("enter the elements of an array \n");
chomp(@arr=<STDIN>);
@reversedarr=reverse(@arr);
foreach(@reversedarr){
print;
}
```

```
+-root@localhost> ./hello.pl
enter the elements of an array
Omnia
rana
hana
hanaranaOmnia
```

8)Write a program that reads a number and then a list of strings (all on separate lines), and then prints one of the lines from the list as selected by the number.

```
#! /usr/bin/perl -w
print("enter the number \n");
chomp($num=<STDIN>);
print("enter the elements of an array \n");
chomp(@arr=<STDIN>);
$item=$arr[$num];
print("element no $num in the array is:$item");
```

```
+-root@localhost> ./hello.pl
enter the number
1
enter the elements of an array
omnia
hana
rana
element no 1 in the array is:hana
```

9)Write a program that reads a list of strings and then selects and prints a random string from the list.

```
#! /usr/bin/perl -w
print("enter a list of strings  \n");
chomp(@arr=<STDIN>);
$item=$arr[rand(@arr)];
print("a random string from the list:$item");
```

```
+-root@localhost> ./hello.pl
enter a list of strings
Omnia
hana
rana
a random string from the list:rana
```

10)Write a program that asks for the temperature outside, and prints "too hot" if the temperature is above 35, and "too cold" otherwise.

```
#! /usr/bin/perl -w
print("enter the the temperature outside \n");
chomp($temp=<STDIN>);
if($temp>35){
print("too hot");
}else{
print("too cold");}
```

```
,---(13:25:35)--[/root]
+-root@localhost> ./hello.pl
enter the the temperature outside
39
too hot
,---(13:31:33)--[/root]
+-root@localhost> ./hello.pl
enter the the temperature outside
20
too cold
```

11) Modify the program from the previous exercise so that it prints "too hot" if the temperature is above 35, "too cold" if the temperature is below 28, and "just right!" if it is between 28 and 35.

```
#! /usr/bin/perl -w
print("enter the the temperature outside \n");
chomp($temp=<STDIN>);
if($temp>35){
print("too hot");
}elsif($temp<28){
print("too cold");}
else{
print("just right!");}
```

```
,---(13:31:40)--[/root]
+-root@localhost> ./hello.pl
enter the the temperature outside
31
just right!
,---(13:37:03)--[/root]
+-root@localhost> ./hello.pl
enter the the temperature outside
21
too cold
,---(13:37:09)--[/root]
+-root@localhost> ./hello.pl
enter the the temperature outside
40
too hot
```

12) Write a program that reads a list of numbers (on separate lines) until the number 999 is read, and then prints the total of all the numbers added together. (Be sure not to add in the 999!) For example, if you enter 1, 2, 3, and 999, the program should reply with the answer of 6 (1+2+3).

```
#! /usr/bin/perl -w
print("enter the number (enter 999 to exit) \n");
chomp($num=<STDIN>);
while($num !=999){
$add+=$num;
print("enter the number (enter 999 to exit) \n");
chomp($num=<STDIN>);
}
print("the result is $add");
```

```
,---(14:03:50)--[/root]
+-root@localhost> ./hello.pl
enter the number (enter 999 to exit)
1
enter the number (enter 999 to exit)
2
enter the number (enter 999 to exit)
3
enter the number (enter 999 to exit)
999
the result is 6
```

13)Write a program that reads in a list of strings on separate lines and then prints out the list of strings in reverse order - without using reverse on the list.

```
#! /usr/bin/perl -w
print("enter a list of strings  \n");
chomp(@arr=<STDIN>);
$len=@arr;
print("list of strings in reverse order is \n");
for($i=$len-1;$i>=0;$i--){
print("$arr[$i] \n");}
```

```
,---(14:23:00)--[/root]
+-root@localhost> ./hello.pl
enter a list of strings
hana
rana
omnia
sara
list of strings in reverse order is
sara
omnia
rana
hana
```

14) Write a program that prints a table of numbers and their squares from zero to 32. Try to come up with a way where you don't need to have all the numbers from 0 to 32 in a list, and then try one where you do. (For nice looking output, printf "%5g %8g\n", $a, $b prints $a as a five-column number and $b as an eight-column number.)

```
#! /usr/bin/perl -w
print"a table of numbers and their squares from zero to 32.\n";
for($num=0;$num<=32;$num++){
$square=$num*$num;
printf "%5g %8g\n", $num ,$square;}
```

```
a table of numbers and their squares from zero to 32.
    0        0
    1        1
    2        4
    3        9
    4       16
    5       25
    6       36
    7       49
    8       64
    9       81
   10      100
   11      121
   12      144
   13      169
   14      196
   15      225
   16      256
   17      289
   18      324
   19      361
   20      400
   21      441
   22      484
   23      529
   24      576
   25      625
   26      676
   27      729
   28      784
   29      841
   30      900
   31      961
   32     1024
```

15) Write a program that reads in a string, and then prints that string and its mapped value according to the mapping presented in the following table:

Input Output

red apple

green leaves

blue ocean

```
#! /usr/bin/perl -w
%hash=("red","apple","green","leaves","blue","ocean");
print("enter the key \n");
chomp($key=<STDIN>);
print "at $key we have $hash{$key}\n";
```

```
,---(14:30:04)--[/root]
+-root@localhost> ./hello.pl
enter the key
red
at red we have apple

,---(14:51:08)--[/root]
+-root@localhost> ./hello.pl
enter the key
green
at green we have leaves

,---(14:51:20)--[/root]
+-root@localhost> ./hello.pl
enter the key
blue
at blue we have ocean
```

16) Write a program that reads a series of words with one word per line until end-of-file, then prints a summary of how many times each word was seen. (For extra challenge, sort the words in ascending ASCII order in the output.)

```
#! /usr/bin/perl -w
print("enter the words with one word per line \n");
chomp(@arr=<STDIN>);
foreach $i(@arr){
$num{$i}=$num{$i}+1;}
foreach $i(sort keys %num){
print "word $i was seen $num{$i} \n";}
```

```
,---(15:23:12)--[/root]
+-root@localhost> ./hello.pl
enter the words with one word per line
omnia
hana
rana
omnia
hana
word hana was seen 2
word omnia was seen 2
word rana was seen 1
```

17)Write a program “myls” to list the content of a current directory

```
#! /usr/bin/perl -w
@files = glob('*');
foreach $i (@files) {
    print "$i\n";
}
```

```
+-root@localhost> ./hello.pl
anaconda-ks.cfg
CDS.log
CDS.log.cdslck
Desktop
dir1
docs
file1.txt
file1.txt~
file2.txt
hello.pl
hello.pl~
install.log
install.log.syslog
lab
lab.txt
lab_5
lab_tar.tar
libManager.log
oldpasswd
panic.log
passwd
q1.txt.bz2
simulation
tmp
```

18)Write a program to check users login and passwords.

```
#! /usr/bin/perl -w
$name1="omnia";
$pass1="12345";
print("enter your name\n");
chomp($name2=<STDIN>);
chomp($name2);
print("enter your password\n");
chomp($pass2=<STDIN>);
if($name2 eq $name1 && $pass2 eq $pass1){
print"success";
}
else{
print"failed login";}
```

```
,---(15:45:43)--[/root]
+-root@localhost> ./hello.pl
enter your name
omnia
enter your password
12345
success
,---(15:46:07)--[/root]
+-root@localhost> ./hello.pl
enter your name
rana
enter your password
1234
failed login
```

19)Write a program to print the date

Example Tues Apr 19 14:40:00 2011

```
print "the current time is ".(localtime);
```

```
the current time is Wed Nov  8 16:06:58 2023
```