HW 5

Q1) Write a program that takes two strings or numbers on the command line and prints out whether they are equal or not.

My Code:

```perl
#!/usr/bin/env perl
use strict;
use warnings;
use v5.30;
use Scalar::Util qw(looks_like_number);
my ($a, $b) =  @ARGV;


if ($a eq $b)
{
    print "Equal \n";
}
elsif (looks_like_number($a)==1 and looks_like_number($b)==1)
{
    if ($a == $b)
    {
        print "Equal \n";
    }
}
else
{
    print "Not Equal \n";
}
```

Output:

```perl
2   #!/usr/bin/env perl
3   use strict;
4   use warnings;
5   use v5.30;
6   use Scalar::Util qw(looks_like_number);
7   my ($a, $b) =  @ARGV;
8
9
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

deepruparel@Deeps-MacBook-Pro-2 hw5 % perl q1.perl 1 1.0
Equal
deepruparel@Deeps-MacBook-Pro-2 hw5 % perl q1.perl 1 0.1e1
Equal
deepruparel@Deeps-MacBook-Pro-2 hw5 % perl q1.perl windy Windy
Not Equal
deepruparel@Deeps-MacBook-Pro-2 hw5 %
```

The equal operator in perl eq is used to compare just strings and only compares them. But our program requires us to compare numbers as well, so we use the looks_like_number which checks if the input is number and then runs the comparison operator == to check if they are equal. When comparing windy and Windy, it is not equal because windy is lowercase while Windy starts with an uppercase letter followed by lowercase letters.

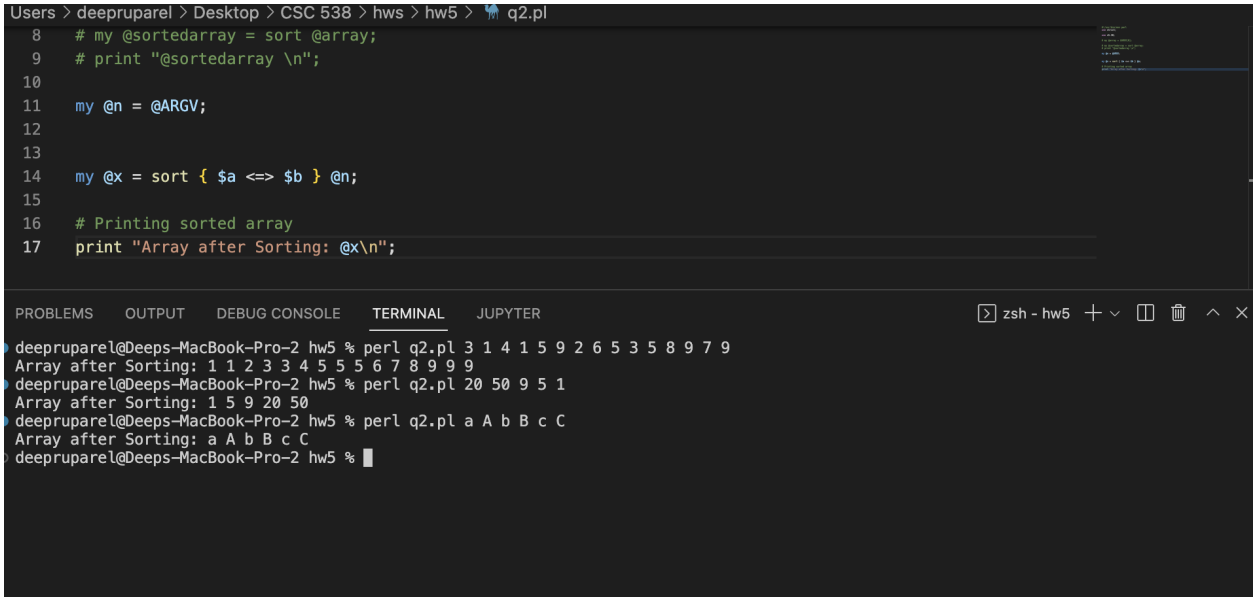
Q2) Write a program to sort a list of numbers from the command line and print them in ascending order.

My Code:

Users > deepruparel > Desktop > CSC 538 > hws > hw5 > q2.pl

```
1   #!/usr/bin/env perl
2   use strict;
3
4   use v5.30;
5
6   # my @array = $ARGV[0];
7
8   # my @sortedarray = sort @array;
9   # print "@sortedarray \n";
10
11  my @n = @ARGV;
12
13
14  my @x = sort { $a <=> $b } @n;
15
16  # Printing sorted array
17  print "Array after Sorting: @x\n";
```

Output:



The Perl sort() function sorts strings as characters but we need to sort it numerically rather than alphabetically. In order to get the input sorted numerically I can make use of the spaceship operator which compares the elements numerically.
For the last input, my program doesn't sort the input because the input is string and the spaceship operator just sorts it numerically so the input is not sorted rather just returned as it was.