

```
use strict;  
use warnings;  
  
print "Hello world!\n";
```

```
use strict;  
use warnings;  
  
my $x = 5;  
print "x = $x\n";  
  
$x = "five";  
print "x = $x\n";  
  
$x = 5;  
my $y = 8;  
my $z = $x + $y;  
  
print "$x + $y = $z\n";
```

```
use strict;
use warnings;

my @n = (1,2,3,4,5);
print "@n\n";

print "$n[2]\n";

my $element = $n[2];
print "$element\n";

my $index = 4;
print "$n[$index]\n";

print "$n[-1]\n";

my $no_elements = @n;
print "$no_elements\n";
```

```
use strict;
use warnings;

my @n = (1,2,3,4,5);
foreach (@n) {
    print "$_\n";
}

foreach my $element (@n) {
    print "$element\n";
}

for (my $i = 0; $i < @n ; $i++) {
    print "$i: $n[$i]\n";
}
```

```
use strict;
use warnings;

my @n = (1,2,3,4,5);
foreach (@n) {
    if ($_ == 4) {
        print "The array contains a 4\n";
    }
}

@n = (1,2,3,4,5);
foreach (@n) {
    if ($_ == 4) {
        print "The array contains a 4\n";
        last;
    }
}
```

```
use strict;
use warnings;

my ($a,$b,$c) = @ARGV;

my $root = $b**2-4*$a*$c;

if ($root < 0) {
    print "No solution!\n";
    exit;
}

my $x1 = (-$b + sqrt($root))/(2*$a);

my $x2 = (-$b - sqrt($root))/(2*$a);

if ($x1 == $x2) {
    print "Solution: $x1\n";
} else {
    print "Solutions: $x1 and $x2\n";
}
```

```
use strict;  
use warnings;  
  
my @list = (1,4,7,8);  
my $c = 10;  
  
for (my $i = 0; $i < @list; $i++) {  
    $list[$i] += $c;  
}  
  
print "@list\n";
```

```
use strict;
use warnings;

my @list = @ARGV;

my @newlist;

foreach my $l (@list) {

    my $found = 0;
    foreach my $n (@newlist) {
        if ($l eq $n) {
            $found = 1;
            last;
        }
    }

    #print "$l $found\n";

    if (!$found) {
        push @newlist, $l;
    }
}

print "@newlist\n";
```



```
use strict;
use warnings;

my @n = @ARGV;

my @c;
foreach (@n) {
    push @c, 0;
}

my $no_comb = 0;

while (1) {

    print "@c\n";
    $no_comb++;

    for (my $i = $#n; $i >= 0; $i--) {
        if ($c[$i] == $n[$i]) {
            $c[$i] = 0;
        } else {
            $c[$i]++;
            last;
        }
    }

    # Back to 0,0,0, ...
    my $back = 1;
    foreach (@c) {
        if ($_ != 0) {
            $back = 0; last;
        }
    }
    last if $back;
}

print "No combinations: $no_comb\n";
```

```
use strict;
use warnings;

my @numbers;

# Opening file ...
open (N, "numbers.txt") || die "Couldn't open file: $!\n";
while (<N>) {
    chomp;

    my @n = split /\s/;
    push @numbers, @n;

    # Or even shorter:
    #push @numbers, split /\s/;
}

print "@numbers\n";

my $sum = 0;
foreach (@numbers) {
    $sum += $_;
}
my $avg = $sum/@numbers;

print "Average = $avg\n";
```

```
use strict;
use warnings;

my @array = (7,92,87,1,4,3,2,6);

my $n = @array;
for (my $i = 0; $i <= $n-2; $i++) {

    my $index = $i;
    for (my $k = $i+1; $k < $n; $k++) {
        if ($array[$k] < $array[$index]) {
            $index = $k;
        }
    }

    # print "$i $index: @array\n";

    ($array[$i], $array[$index]) = ($array[$index], $array[$i]);
}

print "@array\n";
```