```
%Number 1
format short e;
x = .1
```

```
x =
   1.0000e-01
```

```
format long e;
x = .1
```

```
x =
     1.000000000000000e-01
```

```
fprintf('%.*e\n', 16, .1)
```

```
1.0000000000000001e-01
```

```
% When d = 16, the 0.1 is represented
% with an extra 1
```

```
% Number 2
uint_to_digits(12345)
```

```
ans = 1×5
     1     2     3     4     5
```

```
uint_to_digits(2^53)
```

```
ans = 1×16
     9     0     0     7     1     9     9     2     5     4     7     4     0 ···
```

```
uint_to_digits(2^53 + 1)
```

```
ans = 1×16
     9     0     0     7     1     9     9     2     5     4     7     4     0 ···
```

```
uint_to_digits(2^53 + 2)
```

```
ans = 1×16
     9     0     0     7     1     9     9     2     5     4     7     4     0 ···
```

```
%Numer 3
format long e;
  digits_to_uint([9 0 0 7 1 9 9 2 5 4 7 4 0 9 9 2])
```

```
ans =
     9.007199254740992e+15
```

```
  digits_to_uint([9 0 0 7 1 9 9 2 5 4 7 4 0 9 9 3])
```

```
ans =
     9.007199254740992e+15
```

```
  digits_to_uint([9 0 0 7 1 9 9 2 5 4 7 4 0 9 9 4])
```

```
ans =
     9.007199254740994e+15
```

```
% Number 4
addn([2 3 4 6], [9 9 9 9])
```

```
ans =
       12345
```

```
% Number 5
multn([8 2 3],[1 5])
```

```
totalNumber =
        8000
totalNumber =
       12000
totalNumber =
       12200
totalNumber =
       12300
totalNumber =
       12330
totalNumber =
       12345
ans =
       12345
```

```
function X = uint_to_digits(x)  %Function 1
    numDigits = 0;
    numDigits = floor(log10(x)) + 1; %4 + 1
    arr = zeros(1,numDigits);

    for i = numDigits:-1:1 %we're getting a specific digit place with floor(mod(x,10^i
        digitPlace = floor(mod(x,10^i)/10^(i-1)) ;
        arr(1,numDigits-i+1) = digitPlace;
    end
    X = arr;
end
```

```
%Function 2
function x = digits_to_uint(X)
    totalNumber = 0;
    [m,x] = size(X);

    for (i = 1:x) %we're getting a specific digit place with (X(1,x-i+1) * 10^(i-1))
        totalNumber = totalNumber + (X(1,x-i+1) * 10^(i-1));
    end
    x = totalNumber;
end

%Function 3
function s = addn(a, b)
    [m,aSize] = size(a);
    [m,bSize] = size(b);

    totalNumber = 0;

    for (i=1:aSize) %i = places
        totalNumber = totalNumber + ((a(1,aSize-i+1) + b(1,bSize-i+1)) *10^(i-1));
    end
    s = totalNumber;
end

%Function 4
function s = multn(a,b)
    [m,aSize] = size(a);
    [m,bSize] = size(b);
    totalNumber = 0;

    for (i = 1: aSize)
        for (y=1:bSize)
            totalNumber = totalNumber + (a(1,i)*10^(aSize-i)) * b(1,y)*10^(bSize-y)
        end
    end
    s = totalNumber;
end
```