```
def all_divisible_nums (num, divisar)

  return [] if num < divisar
  return [num] if num == divisar

  if (num % divisar).zero?
    all_divisible_nums (num-1, divisor) << num
  else
    all_divisible_nums (num-1, divisar)
  end
end
```

all_divisible_nums(6, 3)

[3, 6]

```
def all_divisible_nums (num, divisar)

  return [] if num < divisar
  return [num] if num == divisar

  if (num % divisar).zero?
    all_divisible_nums (num-1, divisor) << num
  else
    all_divisible_nums (num-1, divisar)
  end
end
```

all_divisible_nums(5, 3)

[3]

```
def all_divisible_nums (num, divisar)

  return [] if num < divisar
  return [num] if num == divisar

  if (num % divisar).zero?
    all_divisible_nums (num-1, divisor) << num
```

all_divisible_nums(4, 3)

```
  else
    all_divisible_nums(num-1, divisor)
    [3]
  end
end

def all_divisible_nums(num, divisar)                all_divisible_nums(3, 3)

  return [] if num < divisar
  return [num] if num == divisar

  if (num % divisar).zero?
    all_divisible_nums(num-1, divisor) << num
  else
    all_divisible_nums(num-1, divisor)
  end
end
```