

```

func is_prime(n int) bool {
    if n < 2 {
        return false
    } else if n == 2 {
        return true
    } else if n % 2 == 0 {
        return false
    } else {
        candidate := 3
        for candidate * candidate <= n {
            if n % candidate == 0 {
                return false
            }
            candidate += 2
        }
        return true
    }
}

```

Go

```

def is_prime?(n)
    return false if n < 2
    return true if n == 2
    return false if n.even?

    candidate = 3
    while candidate * candidate <= n
        return false if n % candidate == 0
        candidate += 2
    end
    return true
end

```

Ruby

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Testing for Primes

```
func std_dev(nums []float64) float64 {
    avg := mean(nums)
    result := 0.0
    for _, x := range nums {
        diff := x - avg
        result += diff * diff
    }
    return math.Sqrt(result / float64(len(nums)))
}
```

mean is a function
written by the
programmer

```
def std_dev(nums)
```

```
    n = nums.length
```

```
    mean = nums.sum / n
```

```
    diffs = nums.map { |x| (x - mean)**2 }
```

```
    return Math.sqrt(diffs.sum / n)
```

```
end
```

Go

Ruby

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Standard Deviation

```
func nbits(n int) []string {
```

```
    if n < 0 {
```

```
        return []string{}
```

Go

```
    } else if n == 1 {
```

```
        return []string{"0", "1"}
```

```
    } else {
```

```
        n1bits := nbits(n - 1)
```

```
        zero := append([]string{}, n1bits...)
```

```
        one := append([]string{}, n1bits...)
```

```
        for i := range n1bits {
```

```
            zero[i] = "0" + zero[i]
```

```
            one[i] = "1" + one[i]
```

```
        }
```

```
        return append(zero, one...)
```

```
    }
```

```
}
```

```
def nbits(n)
```

```
    return [] if n < 0
```

```
    return ['0','1'] if n == 1
```

Ruby

```
    n1bits = nbits(n-1)
```

```
    zero = n1bits.map {|s| '0' + s}
```

```
    one = n1bits.map {|s| '1' + s}
```

```
    return zero + one
```

```
end
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Generating Bits