

```
import random

def fermat_test(n, k=5):
    if n <= 1:
        return False
    if n <= 3:
        return True
    for _ in range(k):
        a = random.randint(2, n - 2)
        if pow(a, n - 1, n) != 1:
            return False
    return True

# Пример использования:
n = 29
print(f"Число {n} вероятно простое: {fermat_test(n)}")
```

➞ Число 29 вероятно простое: True

```
import random

def jacobi_symbol(a, n):
    if n <= 0 or n % 2 == 0:
        return 0
    a = a % n
    result = 1
    while a != 0:
        while a % 2 == 0:
            a = a // 2
            if n % 8 in [3, 5]:
                result = -result
        a, n = n, a
        if a % 4 == 3 and n % 4 == 3:
            result = -result
        a = a % n
    if n == 1:
        return result
    else:
        return 0

def solovay_strassen_test(n, k=5):
    if n <= 1:
        return False
    if n <= 3:
        return True
    for _ in range(k):
        a = random.randint(2, n - 2)
        r = pow(a, (n - 1) // 2, n)
        if r != 1 and r != n - 1:
            return False
        s = jacobi_symbol(a, n)
        if r != s % n:
            return False
    return True

# Пример использования:
n = 29
print(f"Число {n} вероятно простое: {solovay_strassen_test(n)}")
```

➞ Число 29 вероятно простое: True

```
import random


def miller_rabin_test(n, k=5):
    if n <= 1:
        return False
    if n <= 3:
        return True
    if n % 2 == 0:
        return False

    # Представляем n - 1 в виде 2^s * r
    s = 0
    r = n - 1
    while r % 2 == 0:
        s += 1
        r //= 2

    for _ in range(k):
        a = random.randint(2, n - 2)
```

```
y = pow(a, r, n)
if y != 1 and y != n - 1:
    j = 1
    while j < s and y != n - 1:
        y = pow(y, 2, n)
        if y == 1:
            return False
        j += 1
    if y != n - 1:
        return False
return True

# Пример использования:
n = 29
print(f"Число {n} вероятно простое: {miller_rabin_test(n)}")
```

 Число 29 вероятно простое: True