def kiemtra\_nuocsoi(nhiet\_do):

    return 'Nuoc ' + ('da soi!' if nhiet\_do == 100 else 'chua soi!')

print(kiemtra\_nuocsoi(100))

print(kiemtra\_nuocsoi(99))

Kết quả:

Nuoc da soi!

Nuoc chua soi!

for diem\_so in range(10):

    ds\_dau = [str(diem\_so) for diem\_so in range(10) if diem\_so > 5]

    if diem\_so > 5:

        ds\_dau.append(str(diem\_so))

print(ds\_dau)

Kết quả:

['6', '7', '8', '9', '9']

a = True

b = False

1/0

if (a == b) and (1/0 > 0):

    print("a=b")

else:

    print("a khác b")

Kết quả:

Câu lệnh if trong đoạn mã không được thực hiện vì lỗi ZeroDivisionError (lỗi không chia được cho 0)

a = True

b = False

if (1/0>0) and (a==b):

    print("a=b")

else:

    print("a khác b")

Kết quả:

Do 1/0 là phép chia một số cho 0, nó sẽ dẫn đến lỗi ZeroDivisionError

Do lỗi xảy ra, câu lệnh if sẽ không được thực thi

Do đó, câu lệnh else sẽ được thực thi và in ra "a khác b"

S = [2\*x for x in range(20)]

print(S)

Kết quả:

[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38]

A = set([x for x in range(-50, 50) if x\*\*2 + x - 6 == 0])

B = set([2, -3])

print(A==B)

Kết quả:

True

khoảng (-50, 50) có ý nghĩa gì trong lệnh trên? Thay thế khoảng khác được không? Lí do?

khoảng (-50, 50) có nghĩa là tạo ra một danh sách các số nguyên từ -50 đến 49 (không bao gồm 50).

Có thể thay thế khoảng (-50, 50) bằng một khoảng khác.

Lý do là biểu thức list comprehension chỉ sử dụng các giá trị trong khoảng để tạo danh sách.

Các giá trị trong danh sách này được sử dụng để tạo tập hợp A.

def isPrime(N):

    for i in [ x+1 for x in range(N) ]:

            if N % i == 0 and (i!=1 and i!=N):

                return False

    return True

S\_prime = set([x for x in range(40) if (isPrime(x) == True and x>0)])

print(S\_prime)

Kết quả:

{1, 2, 3, 5, 37, 7, 11, 13, 17, 19, 23, 29, 31}

Apro = set([2\*x \*y for x in range(-50,50) if x\*\*2+x-6 == 0 for y in [1, 10, 100]])

print(Apro)

Kết quả:

{-60, 4, -600, 40, 400, -6}

AB = set([ (x, 2\*y) for x in range(-50,50) if x\*\*2+x-6 == 0 for y in [1, 10, 100]])

print(AB)

Kết quả:

{(2, 20), (-3, 20), (-3, 200), (2, 200), (2, 2), (-3, 2)}

from sympy import FiniteSet

phantu = [2, 4, 6, 8, 10]

tap = FiniteSet(\*phantu)

print(tap)

Kết quả:

{2, 4, 6, 8, 10}

from sympy import FiniteSet

phantu = [6, 7, 8, 9, 6, 7]

taphop = FiniteSet(\*phantu)

print(taphop)

for thanhphan in taphop:

    print(thanhphan)

Kết quả:

{6, 7, 8, 9}

6

7

8

9

from sympy import FiniteSet

ds1 = [2, 4, 6]

ds2 = [6, 2, 4]

s = FiniteSet(\*ds1)

t = FiniteSet(\*ds2)

print(ds1 == ds2)

print(s == t)

Kết quả:

False

True

from sympy import FiniteSet

s = FiniteSet(1)

t = FiniteSet(1,2)

print(s.is\_subset(t))

print(t.is\_subset(s))

print(t.is\_subset(t))

print(s.is\_superset(t))

print(t.is\_superset(s))

Kết quả:

True

False

True

False

True