The problem of deciding if a program will output the number 420 is computable

- ○ True
- ◉ False

Check420(P, x)

The problem of deciding if a program will print the number 420 in the first 420 steps of its execution is computable

- ◉ True
- ○ False

1

```
def TestHalt(P, x):
    def P'(x):
        P(x) suppress output
        return 420
    if Check420(P', x):
        return "halts"
    else:
        return "loops"
```

t	l
1	1
2	3
3	1
4	2

```
1 def foo():
2     print("bar")
3 foo()
```

2

Turing

Turing(Turing)

Case I: Turing term => Turing loops

Case II:

HW2, Q3

CheckFix(F)

```
def TestHalt(F, x):
    def F'(y):
        F(x)
        return y
    if CheckFix(F'):
        return "halts"
    else:
        return "loops"
```

contra

assume computable

$f(x) = x$

$f(x) = x^2$

0, 1

$f(0) = 0^2 = 0$

$f(1) = 1^2 = 1$

$f(x) = x - 1$

CheckFix(F') == true

$\exists y : F'(y) = y$

$\forall y : F'(y) \neq y$

$\neg\exists y$

$x \equiv 5 \pmod 7$
$x \equiv 1 \pmod 2$
$x \equiv 2 \pmod 3$

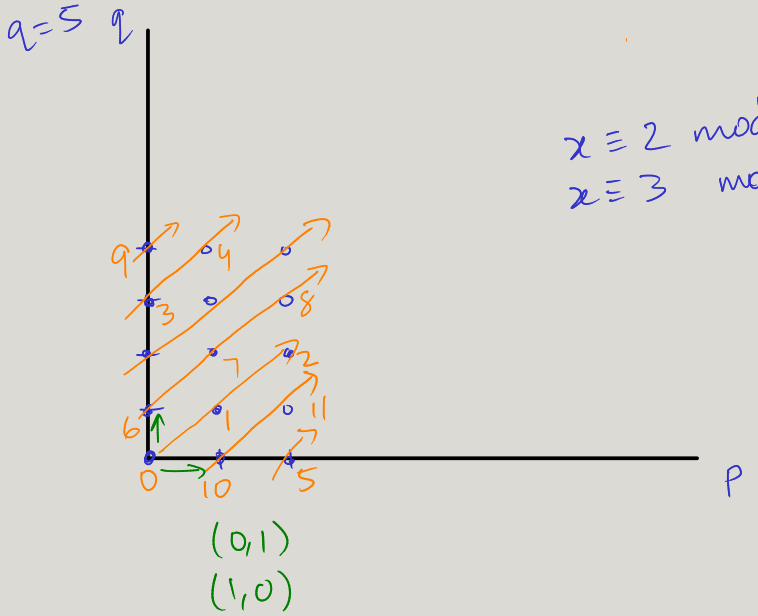
$x \equiv a_1 \pmod{m_1}$
$\vdots$
$x \equiv a_n \pmod{m_n}$
m_i are coprime

5, ~~12~~, 19, ~~26~~, 33, ~~40~~

$x \equiv 2 \pmod 4$
$x \equiv 6 \pmod 8$

2, 6, 10, 14, 18, 22, 26, 30

$$
\begin{aligned}
x \bmod n_i &= ((\sum_{i=1}^{k} a_i b_i) \bmod N) \bmod n_i \\
&= (\sum_{i=1}^{k} a_i b_i) \bmod n_i \\
&= a_i b_i \bmod n_i \\
&= a_i (\frac{N}{n_i} (\frac{N}{n_i})^{-1}_{n_i}) \bmod n_i \\
&= a_i \bmod n_i
\end{aligned}
$$



$x \equiv 2 \bmod 3$
$x \equiv 3 \bmod 5$

mod 15
mod 3
mod 5
0
1 = (1, 1)
2
3
4 = (1, 4)
1+4=5 (2, 0)

p=3
(2, 3)
(0, 4)
(2, 2)
8+9 = 17 = 2 mod 15
2 (2, 3)
2x 8 = 16
1
(4, 6)
(1, 1) = 1