Function Examples

Announcements

cs61a.org/proj/hog_contest

• Up to two people submit one entry; Max of one entry per person

cs61a.org/proj/hog_contest

- Up to two people submit one entry; Max of one entry per person
- Your score is the number of entries against which you win more than 50.00001% of the time

- Up to two people submit one entry; Max of one entry per person
- Your score is the number of entries against which you win more than 50.00001% of the time
- Strategies are time-limited

- Up to two people submit one entry; Max of one entry per person
- Your score is the number of entries against which you win more than 50.00001% of the time
- Strategies are time-limited
- All strategies must be deterministic, pure functions of the players' scores

- Up to two people submit one entry; Max of one entry per person
- Your score is the number of entries against which you win more than 50.00001% of the time
- Strategies are time-limited
- All strategies must be deterministic, pure functions of the players' scores
- Winning entries will receive a paltry amount of extra credit

- Up to two people submit one entry; Max of one entry per person
- Your score is the number of entries against which you win more than 50.00001% of the time
- Strategies are time-limited
- All strategies must be deterministic, pure functions of the players' scores
- Winning entries will receive a paltry amount of extra credit
- The real prize: honor and glory

- Up to two people submit one entry; Max of one entry per person
- Your score is the number of entries against which you win more than 50.00001% of the time
- Strategies are time-limited
- All strategies must be deterministic, pure functions of the players' scores
- Winning entries will receive a paltry amount of extra credit
- The real prize: honor and glory
- See website for detailed rules

- Up to two people submit one entry; Max of one entry per person
- Your score is the number of entries against which you win more than 50.00001% of the time
- Strategies are time-limited
- All strategies must be deterministic, pure functions of the players' scores
- Winning entries will receive a paltry amount of extra credit
- The real prize: honor and glory
- See website for detailed rules

Fall 2011 Winners

Keegan Mann Yan Duan & Ziming Li Brian Prike & Zhenghao Qian Parker Schuh & Robert Chatham

- Up to two people submit one entry; Max of one entry per person
- Your score is the number of entries against which you win more than 50.00001% of the time
- Strategies are time-limited
- All strategies must be deterministic, pure functions of the players' scores
- Winning entries will receive a paltry amount of extra credit
- The real prize: honor and glory
- See website for detailed rules

Fall 2011 Winners

Keegan Mann Yan Duan & Ziming Li Brian Prike & Zhenghao Qian Parker Schuh & Robert Chatham

Fall 2012 Winners

Chenyang Yuan Joseph Hui

- Up to two people submit one entry; Max of one entry per person
- Your score is the number of entries against which you win more than 50.00001% of the time
- Strategies are time-limited
- All strategies must be deterministic, pure functions of the players' scores
- Winning entries will receive a paltry amount of extra credit
- The real prize: honor and glory
- See website for detailed rules

Fall 2011 Winners

Keegan Mann Yan Duan & Ziming Li Brian Prike & Zhenghao Qian Parker Schuh & Robert Chatham

Fall 2012 Winners

Chenyang Yuan Joseph Hui

Fall 2013 Winners

Paul Bramsen Sam Kumar & Kangsik Lee Kevin Chen

cs61a.org/proj/hog_contest

- Up to two people submit one entry; Max of one entry per person
- Your score is the number of entries against which you win more than 50.00001% of the time
- Strategies are time-limited
- All strategies must be deterministic, pure functions of the players' scores
- Winning entries will receive a paltry amount of extra credit
- The real prize: honor and glory
- See website for detailed rules

Fall 2011 Winners

Keegan Mann Yan Duan & Ziming Li Brian Prike & Zhenghao Qian Parker Schuh & Robert Chatham

Fall 2012 Winners

Chenyang Yuan Joseph Hui

Fall 2013 Winners

Paul Bramsen Sam Kumar & Kangsik Lee Kevin Chen

Fall 2014 Winners

Alan Tong & Elaine Zhao Zhenyang Zhang Adam Robert Villaflor & Joany Gao Zhen Qin & Dian Chen Zizheng Tai & Yihe Li

cs61a.org/proj/hog_contest

Hog Contest Winners

Spring 2015 Winners

Sinho Chewi & Alexander Nguyen Tran Zhaoxi Li Stella Tao and Yao Ge

Fall 2015 Winners

Micah Carroll & Vasilis Oikonomou Matthew Wu Anthony Yeung and Alexander Dai

Spring 2016 Winners

Michael McDonald and Tianrui Chen Andrei Kassiantchouk Benjamin Krieges

Fall 2016 Winners

Cindy Jin and Sunjoon Lee Anny Patino and Christian Vasquez Asana Choudhury and Jenna Wen Michelle Lee and Nicholas Chew

Your name could be here FOREVER!

Fall 2017 Winners

Alex Yu and Tanmay Khattar James Li Justin Yokota

Spring 2018 Winners

Eric James Michaud Ziyu Dong Xuhui Zhou

Fall 2018 Winners

Rahul Arya Jonathan Bodine Sumer Kohli and Neelesh Ramachandran

Fall 2019 Winners

Currying

def make_adder(n):
 return lambda k: n + k

```
def make_adder(n):
    return lambda k: n + k
```

```
>>> make_adder(2)(3)
5
>>> add(2, 3)
5
```





(Demo)



Curry: Transform a multi-argument function into a single-argument, higher-order function

(Demo)

Decorators

(Demo)

(Demo)

@trace1
def triple(x):
 return 3 * x







is identical to



is identical to

def triple(x):
 return 3 * x
triple = trace1(triple)



 Review

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

from operator import add, mul
def square(x):
 return mul(x, x)

The print function returns None. (separated by spaces) when it is	It also displays its arguments called.	;	
<pre>from operator import add, mul def square(x): return mul(x, x)</pre>	This expression	Evaluates to	Interactive Output

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	
return mul(x, x)	5	5	

Totoroctivo
<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x , x)	5	5	5

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
	print(5)		

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
	print(5)	None	

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
	print(5)	None	5

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

from operator import add, mul def square(x): return mul(x, x)	This expression	Evaluates to	Output
	5	5	5
	print(5)	None	5
	<pre>print(print(5))</pre>		

Tatoractiva

from operator import add, mul def square(x): return mul(x, x)	This expression	Evaluates to	Interactive Output
	5	5	5
	print(5)	None	5
	<pre>print(print(5)) None</pre>		

from operator import add, mul def square(x): return mul(x, x)	This expression	Evaluates to	Interactive Output
	5	5	5
	print(5)	None	5
	<pre>print(print(5)) None</pre>		5 None

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

from operator import add, mul def square(x): return mul(x, x)	This expression	Evaluates to	Output
	5	5	5
	print(5)	None	5
	<pre>print(print(5)) None</pre>	None	5 None

Totoroctivo

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Output
return mul(x, x)	5	5	5
	print(5)	None	5
<pre>def delay(arg): print('delayed') def g(): return arg return g</pre>	print(<u>print(5)</u>) None	None	5 None

Totoroctivo

This expression	Evaluates to	Output
5	5	5
print(5)	None	5
<pre>print(print(5)) None</pre>	None	5 None
none		
	<pre>This expression 5 print(5) print(print(5)) None</pre>	This expressionEvaluates to55print(5)Noneprint(print(5)) NoneNone

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
that arg	<pre>print(print(5))</pre>	None	5 None
<pre>def delay(arg): print('delayed') def g(): return arg return g Names in nested def statements can refer to their enclosing scope</pre>	None		

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
def (delay(arg):	<pre>print(print(5)) None</pre>	None	5 None
<pre>def g(): return arg return g Names in nested def statements can refer to their enclosing scope</pre>	delay(delay)()(6)()		

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
that arg	<pre>print(print(5)) None</pre>	None	5 None
<pre>def (delay(arg): print('delayed') def g(): return arg return g Names in nested def statements can refer to their enclosing scope</pre>	delay(delay)()(6)()		

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
def (delay(arg):	<pre>print(print(5)) None</pre>	None	5 None
def g(): return arg return g	delay(delay)()(6)()		
statements can refer to their enclosing scope			

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
def (delay(arg): print('delayed')	<pre>print(print(5)) None</pre>	None	5 None
def g(): return (arg) return g	delay(delay)()(6)()		
Names in nested def statements can refer to their enclosing scope			

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
def (delay(arg):	<pre>print(print(5)) None</pre>	None	5 None
<pre>print('delayed') def g(): return arg return g</pre>	(delay(delay)()(6)()		
Names in nested def statements can refer to their enclosing scope			

Totoroctivo

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
that arg	<pre>print(print(5)) None</pre>	None	5 None
<pre>def (delay(arg)): print('delayed') def g(): return (arg) return g</pre>	<pre>delay(delay)()(6)()</pre>		delayed
Names in nested def statements can refer to their enclosing scope			

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
that arg	<pre>print(print(5)) None</pre>	None	5 None
<pre>def (delay(arg): print('delayed') def g(): return (arg) return g</pre>	<pre>delay(delay)()(6)()</pre>		delayed delayed
Names in nested def statements can refer to their enclosing scope			

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
that arg	<pre>print(print(5)) None</pre>	None	5 None
<pre>def (delay(arg)): print('delayed') def g(): return (arg) return g</pre>	<pre>delay(delay)()(6)()</pre>		delayed delayed 6
Names in nested def statements can refer to their enclosing scope			

Totoroativo

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
that arg	<pre>print(print(5)) None</pre>	None	5 None
<pre>def (delay(arg)): print('delayed') def g(): return (arg) return g Names in nested def</pre>	<pre>delay(delay)()(6)()</pre>	6	delayed delayed 6
statements can refer to their enclosing scope			

Totoroativo

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
that arg	<pre>print(print(5)) None</pre>	None	5 None
<pre>def (delay(arg)): print('delayed') def g(): return (arg) return g</pre>	(delay(delay)()()(6)()	6	delayed delayed 6
Names in nested def statements can refer to their enclosing scope	<pre>print(delay(print)()(4))</pre>		

Totoroctivo

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
that arg	<pre>print(print(5)) None</pre>	None	5 None
<pre>def (delay(arg): print('delayed') def g(): return (arg) return g</pre>	<pre>delay(delay)()(6)()</pre>	6	delayed delayed 6
Names in nested def statements can refer to their enclosing scope	<pre>print(delay(print)()(4))</pre>		delayed

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
that arg	<pre>print(print(5)) None</pre>	None	5 None
<pre>def (delay(arg): print('delayed') def g(): return (arg) return g</pre>	(delay(delay))()(6)()	6	delayed delayed 6
Names in nested def statements can refer to their enclosing scope	<pre>print(delay(print)()(4))</pre>		delayed 4

<pre>from operator import add, mul def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
that arg	<pre>print(print(5)) None</pre>	None	5 None
<pre>def (delay(arg)): print('delayed') def g(): return arg return q</pre>	(delay(delay)()(6)()	6	delayed delayed 6
Names in nested def statements can refer to their enclosing scope	<pre>print(delay(print)()(4))</pre>		delayed 4 None

<pre>from operator import add, mu def square(x):</pre>	This expression	Evaluates to	Interactive Output
return mul(x, x)	5	5	5
A function that takes any argument and returns a	print(5)	None	5
that arg	<pre>print(print(5)) None</pre>	None	5 None
<pre>def (delay(arg)): print('delayed') def g(): return (arg) return q</pre>	<pre>delay(delay)()(6)()</pre>	6	delayed delayed 6
Names in nested def statements can refer to their enclosing scope	<pre>print(delay(print)()(4))</pre>	None	delayed 4 None

return mul(x, x)

The print function returns None. It also displays its arguments (separated by spaces) when it is called. from operator import add, mul def square(x): $\frac{This expression}{From operator} = \frac{Evaluates to}{From operator} = \frac{Evaluates$

The print function returns None. It also displays its arguments (separated by spaces) when it is called. from operator import add, mul def square(x): This expression **Evaluates to Interactive Output**

def pirate(arggg):
 print('matey')
 def plunder(arggg):
 return arggg
 return plunder

return mul(x, x)

The print function returns None. (separated by spaces) when it is	It also displays its arguments called.	5	
from operator import add, mul	This expression	Evaluates to	Interactive Output
<pre>def square(x): return mul(x, x)</pre>			

add(pirate(3)(square)(4), 1)

```
def pirate(arggg):
    print('matey')
    def plunder(arggg):
        return arggg
    return plunder
```

The print function returns None. (separated by spaces) when it is	It also displays its arguments called.		
from operator import add, mul	This expression	Evaluates to	Interactive Output
return mul(x, x)			

add(pirate(3)(square)(4), 1)

```
def pirate(arggg):
    print('matey')
    def plunder(arggg):
        return arggg
    return plunder
```

			Interactive
from operator import add, mul	This expression	Evaluates to	Output
<pre>def square(x):</pre>			
return mul(x, x)			
	add(pirate(3)(square)(4), 1)		
A function that			
always returns the			
identity function			
def (pirate(arggg)):			
print('matev')			
<pre>def plunder(arggg):</pre>			
return arggg			
return plunder			

Evaluates to	Output

The print function returns None. It also displays its arguments (separated by spaces) when it is called. Interactive This expression Evaluates to Output from operator import add, mul def square(x): return mul(x, x) Matey add(pirate(3)(square)(4), 1) A function that always returns the identity function def pirate(arggg): print('matey') def plunder(arggg): return arggg return plunder

The print function returns None. It also displays its arguments (separated by spaces) when it is called. Interactive This expression Evaluates to Output from operator import add, mul def square(x): return mul(x, x) Matey add(pirate(3)(square)(4), 1) A function that always returns the identity function def pirate(arggg): print('matey') def plunder(arggg): return arggg return plunder

The print function returns None. It also displays its arguments (separated by spaces) when it is called. Interactive This expression Evaluates to Output from operator import add, mul def square(x): return mul(x, x) Matey add(pirate(3)(square)(4), 1) A function that func square(x) always returns the identity function def pirate(arggg): print('matey') def plunder(arggg): return arggg return plunder

The print function returns None. It also displays its arguments (separated by spaces) when it is called. Interactive This expression Evaluates to Output from operator import add, mul def square(x): return mul(x, x) Matey add(pirate(3)(square)(4), 1) A function that func square(x) always returns the identity function def pirate(arggg): print('matey') def plunder(arggg): return arggg return plunder

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x);</pre>	This expression	Evaluates to	Output
<pre>def square(x): return mul(x, x) A function that always returns the</pre>	<pre>add(pirate(3)(square)(4), 1) func square(x)</pre>		Matey
<pre>identity function def pirate(arggg): print('matey') def plunder(arggg): return arggg return plunder</pre>	16		

Interactive
The print function returns None. It also displays its arguments (separated by spaces) when it is called.

der square(x):	
return mul(x, x) add(<u>pirate(3)</u> (square)(4), 1) 17	еу
A function that <i>func square(x)</i> always returns the	
identity function 16	
<pre>def pirate(arggg): print('matey') def plunder(arggg): return arggg return plunder</pre>	

Interactive

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

from operator import add, mul	This expression	Evaluates to	Output
<pre>def square(x): return mul(x, x) A function that always returns the identity function def pirate(arggg): print('matey') def plunder(arggg): return arggg return plunder</pre>	<pre>add(<u>pirate(3)(square)(4), 1) func square(x) 16</u></pre>	17	Matey 17

Interactive

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x);</pre>	This expression	Evaluates to	Output
A function that always returns the identity function	<pre>add(pirate(3)(square)(4), 1) func square(x) </pre>	17	Matey 17
<pre>def (pirate(arggg)): print('matey') def plunder(arggg): return arggg return plunder</pre>	<pre>10 pirate(pirate(pirate))(5)(7)</pre>		

Intoractivo

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

from operator import add, mul	This expression	Evaluates to	Output
return mul(x, x)	<pre>add(pirate(3)(square)(4), 1)</pre>	17	Matey 17
always returns the identity function	func square(x) 		
<pre>def pirate(arggg): print('matey') def plunder(arggg): return arggg return plunder</pre>	<pre>pirate(pirate(pirate))(5)(7)</pre>		

Interactive

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x);</pre>	This expression	Evaluates to	Output
A function that	<pre>add(pirate(3)(square)(4), 1) func square(x)</pre>	17	Matey 17
always returns the identity function	16		
<pre>def pirate(arggg): print('matey') def plunder(arggg): return arggg return plunder</pre>	<pre>pirate(pirate(pirate))(5)(7) Identity function</pre>		

Interactive

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x);</pre>	This expression	Evaluates to	Output
return mul(x, x)	<pre>add(pirate(3)(square)(4), 1)</pre>	17	Matey 17
A function that always returns the identity function	func square(x)		
<pre>def (pirate(arggg)):</pre>	<pre>pirate(pirate(pirate))(5)(7)</pre>		Matey
def plunder(arggg): return arggg return plunder	Identity function		

Interactive

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def cquare(x);</pre>	This expression	Evaluates to	Output
return mul(x, x)	<pre>add(pirate(3)(square)(4), 1)</pre>	17	Matey 17
A function that always returns the	<pre>func square(x)</pre>		17
identity function	16		
<pre>def (pirate(arggg): print('matey')</pre>	<pre>pirate(pirate(pirate))(5)(7)</pre>		Matey Matev
def plunder(arggg): return arggg return plunder	Identity function		

Interactive

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

from operator import add, mul	This expression	Evaluates to	Output
return mul(x, x)	<pre>add(pirate(3)(square)(4), 1)</pre>	17	Matey 17
A function that always returns the	<pre>func square(x)</pre>		1,
identity function	16		
<pre>def (pirate(arggg)): print('matey')</pre>	<pre>pirate(pirate(pirate))(5)(7)</pre>		Matey Matey
def plunder(arggg): return arggg return plunder	Identity function		

Interactive

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x);</pre>	This expression	Evaluates to	Output
A function that always returns the identity function	<pre>add(pirate(3)(square)(4), 1) func square(x) 16</pre>	17	Matey 17
<pre>def(pirate(arggg)): print('matey')</pre>	<pre>pirate(pirate(pirate))(5)(7)</pre>		Matey Matey
def plunder(arggg): return arggg return plunder	Identity function 5		hacey

Intoractivo

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<pre>from operator import add, mul def square(x);</pre>	This expression	Evaluates to	Output
A function that always returns the identity function	<pre>add(<u>pirate(3)(square)(4), 1) func square(x) 16</u></pre>	17	Matey 17
<pre>def (pirate(arggg)): print('matey')</pre>	<pre>pirate(pirate(pirate))(5)(7)</pre>		Matey Matey
<pre>def plunder(arggg): return arggg </pre>	Identity function		Error
recurn plunder	5		

Intoractivo

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

from operator import add, mul	This expression	Evaluates to	Output
A function that always returns the identity function	<pre>add(pirate(3)(square)(4), 1) func square(x) 16</pre>	17	Matey 17
<pre>def (pirate(arggg)): print('matey') def plunder(arggg):</pre>	<pre>pirate(pirate(pirate))(5)(7)</pre>	Error	Matey Matey
return arggg return plunder	Identity function 5		Error

Interactive



















horse(mask)









horse(mask)





horse(mask)





























Global frame

func horse(mask) [parent=Global] horse mask f1: horse [parent=Global] mask •-horse 🚽 func λ(horse) [parent=Global] Return Value f2: λ [parent=Global] func mask(horse) [parent=f1] horse Return Value _____ f3: mask [parent=f1] -Return Value



















Global frame

func horse(mask) [parent=Global] horse mask f1: horse [parent=Global] mask • horse 🚽 func λ(horse) [parent=Global] 2 Return Value f2: λ [parent=Global] func mask(horse) [parent=f1] horse 2 Return Value f3: mask [parent=f1] 2 horse Return Value 2

Implementing Functions

Implementing a Function

```
def remove(n, digit):
   """Return all digits of non-negative N
     that are not DIGIT, for some
     non-negative DIGIT less than 10.
  >>> remove(231, 3)
  21
  >>> remove(243132, 2)
  4313
   .....
  kept, digits = 0, 0
  while
     n, last = n // 10, n % 10
     if _____:
        kept = _____
        digits = _____
  return _____
```

Implementing a Function

```
def remove(n, digit):
   """Return all digits of non-negative N
     that are not DIGIT, for some
     non-negative DIGIT less than 10.
  >>> remove(231, 3)
  21
  >>> remove(243132, 2)
  4313
   .....
   kept, digits = 0, 0
  while
     n, last = n // 10, n % 10
     if _____:
        kept = _____
        digits = _____
  return _____
```

Read the description
```
def remove(n, digit):
   """Return all digits of non-negative N
     that are not DIGIT, for some
     non-negative DIGIT less than 10.
  >>> remove(231, 3)
   21
  >>> remove(243132, 2)
  4313
   .....
   kept, digits = 0, 0
  while :
     n, last = n // 10, n % 10
     if _____:
        kept = _____
        digits = _____
  return _____
```

Read the description

Verify the examples & pick a simple one

```
def remove(n, digit):
                                      Read the description
   """Return all digits of non-negative N
     that are not DIGIT, for some
                                      Verify the examples & pick a simple one
     non-negative DIGIT less than 10.
   >>> remove(231, 3)
                                      Read the template
   21
   >>> remove(243132, 2)
   4313
   .....
   kept, digits = 0, 0
   while _____:
      n, last = n // 10, n % 10
      if _____:
         kept = _____
         digits = _____
   return _____
```

```
def remove(n, digit):
   """Return all digits of non-negative N
     that are not DIGIT, for some
     non-negative DIGIT less than 10.
  >>> remove(231, 3)
   21
  >>> remove(243132, 2)
  4313
   .....
  kept, digits = 0, 0
  while
     n, last = n // 10, n % 10
     if _____ :
        kept =
        digits = _____
  return _____
```

Read the description Verify the examples & pick a simple one Read the template Implement without the template, then change your implementation to match the template. OR If the template is helpful, use it.

```
def remove(n, digit):
   """Return all digits of non-negative N
     that are not DIGIT, for some
     non-negative DIGIT less than 10.
  >>> remove(231, 3)
   21
  >>> remove(243132, 2)
  4313
   .....
  kept, digits = 0, 0
  while
     n, last = n // 10, n % 10
     if _____ :
        kept =
        digits = _____
  return _____
```

Read the description
Verify the examples & pick a simple one
Read the template
Implement without the template, then change
your implementation to match the template.
OR
If the template is helpful, use it.
Annotate names with values from your chosen
example

```
def remove(n, digit):
                                          Read the description
   """Return all digits of non-negative N
     that are not DIGIT, for some
                                          Verify the examples & pick a simple one
      non-negative DIGIT less than 10.
   >>> remove(231, 3)
                                          Read the template
   21
   >>> remove(243132, 2)
                                          Implement without the template, then change
   4313
                                          your implementation to match the template.
   .....
                                          OR
   kept, digits = 0, 0
                                          If the template is helpful, use it.
   while
                                          Annotate names with values from your chosen
                                          example
      n, last = n // 10, n % 10
      if _____
                                          Write code to compute the result
          kept =
          digits = _____
   return _____
```

```
def remove(n, digit):
                                          Read the description
   """Return all digits of non-negative N
     that are not DIGIT, for some
                                          Verify the examples & pick a simple one
      non-negative DIGIT less than 10.
   >>> remove(231, 3)
                                          Read the template
   21
   >>> remove(243132, 2)
                                          Implement without the template, then change
   4313
                                          your implementation to match the template.
   .....
                                          OR
   kept, digits = 0, 0
                                          If the template is helpful, use it.
   while
                                          Annotate names with values from your chosen
                                          example
      n, last = n // 10, n % 10
      if _____
                                          Write code to compute the result
          kept =
                                          Did you really return the right thing?
          digits = _____
   return
```

```
def remove(n, digit):
                                           Read the description
   """Return all digits of non-negative N
      that are not DIGIT, for some
                                           Verify the examples & pick a simple one
      non-negative DIGIT less than 10.
   >>> remove(231, 3)
                                           Read the template
   21
   >>> remove(243132, 2)
                                           Implement without the template, then change
   4313
                                           your implementation to match the template.
   .....
                                           OR
   kept, digits = 0, 0
                                           If the template is helpful, use it.
   while
                                           Annotate names with values from your chosen
                                           example
       n, last = n // 10, n % 10
       if _____
                                           Write code to compute the result
                                           Did you really return the right thing?
          kept =
          digits = _____
                                           Check your solution with the other examples
   return _____
```

```
def remove(n, digit):
    """"Return all digits of non-negative N
                                            Read the description
                3 IT, for some
           ire
      231
                                            Verify the examples & pick a simple one
                    IT less than 10.
           aal
   >>> remove(231, 3)
                                            Read the template
   21
   >>> remove(243132, 2)
                                            Implement without the template, then change
   4313
                                            your implementation to match the template.
   .....
                                            OR
   kept, digits = 0, 0
                                            If the template is helpful, use it.
   while
                                            Annotate names with values from your chosen
                                            example
       n, last = n // 10, n % 10
                                            Write code to compute the result
       if _____
                                            Did you really return the right thing?
           kept =
           digits = ___
                                            Check your solution with the other examples
   return _____
```

```
def remove(n, digit):
    """"Return all digits of non-negative N
                                            Read the description
                3 IT, for some
           ire
      231
                                            Verify the examples & pick a simple one
                    IT less than 10.
           aal
   >>> remove(231, 3)
                                            Read the template
   21
   >>> remove(243132, 2)
                                            Implement without the template, then change
   4313
                                            your implementation to match the template.
   .....
                                            OR
   kept, digits = 0, 0
                                            If the template is helpful, use it.
   while
                                            Annotate names with values from your chosen
                                            example
       n, last = n // 10, n % 10
                                            Write code to compute the result
       if _____
                                            Did you really return the right thing?
           kept =
     21
           digits = _____
                                            Check your solution with the other examples
   return
```

```
def remove(n, digit):
    """"Return all digits of non-negative N
                                              Read the description
                 3 IT, for some
            ire
      231
                                              Verify the examples & pick a simple one
                     IT less than 10.
            aal
   >>> remove(231, 3)
                                              Read the template
   21
   >>> remove(243132, 2)
                                              Implement without the template, then change
   4313
                                              your implementation to match the template.
   .....
                                              OR
   kept, digits = 0, 0
                                              If the template is helpful, use it.
   while ____ n > 0
                                              Annotate names with values from your chosen
                                              example
       n, last = n // 10, n % 10
                                              Write code to compute the result
       if
                                              Did you really return the right thing?
           kept =
     21
           digits = ___
                                              Check your solution with the other examples
    return
```





```
def remove(n, digit):
    """"Return all'digits of non-negative N
                                                   Read the description
                       IT, for some
                   3
             ire
       231
                                                   Verify the examples & pick a simple one
                       IT less than 10.
             eaa
    >>> remove(231, 3)
                                                   Read the template
    21
    >>> remove(243132, 2)
                                                   Implement without the template, then change
    4313
                                                   your implementation to match the template.
    .....
                                                   OR
    kept, digits = 0, 0
                                                   If the template is helpful, use it.
    while n > 0
                                                   Annotate names with values from your chosen
                                                   example
        n, last = n // 10, n % 10
                last != digit
                                                   Write code to compute the result
        if
                      kept + last
                                                   Did you really return the right thing?
            kept =
      21
            digits = ____
                                                   Check your solution with the other examples
                        kept
    return
```

```
def remove(n, digit):
    """"Return all'digits of non-negative N
                                                  Read the description
                       IT, for some
                   3
             ire
       231
                                                  Verify the examples & pick a simple one
                       IT less than 10.
             eaa
    >>> remove(231, 3)
                                                  Read the template
    21
    >>> remove(243132, 2)
                                                   Implement without the template, then change
    4313
                                                  your implementation to match the template.
    .....
                                                  OR
    kept, digits = 0, 0
                                                   If the template is helpful, use it.
    while n > 0
                                                  Annotate names with values from your chosen
                                                   example
        n, last = n // 10, n % 10
                last != digit
                                                  Write code to compute the result
        if
            kept = 10*kept + last
                                                  Did you really return the right thing?
      21
            digits = _____
                                                  Check your solution with the other examples
                        kept
    return
```

```
def remove(n, digit):
    ali diaits of non-negative N
                                                 Read the description
                       IT, for some
                  3
             re
       231
                                                 Verify the examples & pick a simple one
                       IT less than 10.
             eaa
    >>> remove(231, 3)
                                                 Read the template
    21
    >>> remove(243132, 2)
                                                 Implement without the template, then change
    4313
                                                 your implementation to match the template.
    .....
                                                 OR
    kept, digits = 0, 0
                                                 If the template is helpful, use it.
                   n > 0
    while
                                                 Annotate names with values from your chosen
                                                 example
        n, last = n // 10, n % 10
                last != digit
                                                 Write code to compute the result
        if
                    D⊗*kept + last
                                                 Did you really return the right thing?
            kept =
      21
            digits =
                                                 Check your solution with the other examples
                       kept
    return
```

```
def remove(n, digit):
    ali diaits of non-negative N
                                                 Read the description
                       IT, for some
                  3
             re
       231
                                                 Verify the examples & pick a simple one
                       IT less than 10.
             eaa
    >>> remove(231, 3)
                                                 Read the template
    21
    >>> remove(243132, 2)
                                                 Implement without the template, then change
    4313
                                                 your implementation to match the template.
    .....
                                                 OR
    kept, digits = 0, 0
                                                 If the template is helpful, use it.
                   n > 0
    while
                                                 Annotate names with values from your chosen
                                                 example
        n, last = n // 10, n % 10
                last != digit
                                                 Write code to compute the result
        if
                    LØ*kept + last*10
            kept =
                                                 Did you really return the right thing?
      21
            digits =
                                                 Check your solution with the other examples
                       kept
    return
```

```
def remove(n, digit):
    ali diaits of non-negative N
                                                 Read the description
                       IT, for some
                  Δ
             re
       231
                                                 Verify the examples & pick a simple one
                       IT less than 10.
             eaa
    >>> remove(231, 3)
                                                 Read the template
    21
    >>> remove(243132, 2)
                                                 Implement without the template, then change
    4313
                                                 your implementation to match the template.
    .....
                                                 OR
    kept, digits = 0, 0
                                                 If the template is helpful, use it.
                   n > 0
    while
                                                 Annotate names with values from your chosen
                                                 example
        n, last = n // 10, n % 10
                last != digit
                                                 Write code to compute the result
        if
                    LØ*kept + last*10
            kept =
                                                 Did you really return the right thing?
      21
            digits =
                                                 Check your solution with the other examples
                       kept
    return
```

```
def remove(n, digit):
    """Return alindiaits of non-negative N
                                                  Read the description
                       IT, for some
                  Δ
             re
       231
                                                  Verify the examples & pick a simple one
                       IT less than 10.
             eaa
    >>> remove(231, 3)
                                                  Read the template
    21
    >>> remove(243132, 2)
                                                  Implement without the template, then change
    4313
                                                  your implementation to match the template.
    .....
                                                  OR
    kept, digits = 0, 0
                                                  If the template is helpful, use it.
                   n > 0
    while
                                                  Annotate names with values from your chosen
                                                  example
        n, last = n // 10, n % 10
                last != digit
                                                  Write code to compute the result
        if
                    LØ*kept + last*10
            kept =
                                                  Did you really return the right thing?
     231
            digits =
                                                  Check your solution with the other examples
                        kept
    return
```

```
def remove(n, digit):
    """Return alindiaits of non-negative N
                                                  Read the description
                       IT, for some
                  Δ
             re
       231
                                                  Verify the examples & pick a simple one
                       IT less than 10.
             eaa
    >>> remove(231, 3)
                                                  Read the template
    21
    >>> remove(243132, 2)
                                                  Implement without the template, then change
    4313
                                                  your implementation to match the template.
    .....
                                                  OR
    kept, digits = 0, 0
                                                  If the template is helpful, use it.
                   n > 0
    while
                                                  Annotate names with values from your chosen
                                                  example
        n, last = n // 10, n % 10
                last != digit
                                                  Write code to compute the result
        if
                     D⊗*kept + last*10
                                                  Did you really return the right thing?
            kept =
                       digits + 1
     231
            digits =
                                                  Check your solution with the other examples
                        kept
    return
```

```
def remove(n, digit):
    """Return alindiaits of non-negative N
                                                  Read the description
                       IT, for some
                  Δ
             re
       231
                                                  Verify the examples & pick a simple one
                       IT less than 10.
             eaa
    >>> remove(231, 3)
                                                  Read the template
    21
    >>> remove(243132, 2)
                                                  Implement without the template, then change
    4313
                                                  your implementation to match the template.
    .....
                                                  OR
    kept, digits = 0, 0
                                                  If the template is helpful, use it.
                   n > 0
    while
                                                  Annotate names with values from your chosen
                                                  example
        n, last = n // 10, n % 10
                last != digit
                                                  Write code to compute the result
        if
                    10* kept + last*10**digits
                                                  Did you really return the right thing?
            kept =
                       digits + 1
     231
            digits =
                                                  Check your solution with the other examples
                        kept
    return
```

<pre>def remove(n, digit): """Return all digits of non-negative N TT for some</pre>	Read the description
ga IT less than 10.	Verify the examples & pick a simple one
>>> remove(231, 3) 1 21	Read the template
>>> remove(243132, 2) + 30 4313 + 200	Implement without the template, then change your implementation to match the template.
kept, digits = 0, 0 231	If the template is helpful, use it.
while: n, last = n // 10, n % 10	Annotate names with values from your chosen example
iflast != digit:	Write code to compute the result
kept =kept + last*10**digits	Did you really return the right thing?
231 digits = <u>digits + 1</u>	Check your solution with the other examples
return kept	

```
def remove(n, digit):
    """"Return all'digits of non-negative N
                                                  Read the description
                       IT, for some
                   3
             ire
       231
                                                  Verify the examples & pick a simple one
                       IT less than 10.
             eaa
    >>> remove(231, 3)
                                                  Read the template
    21
    >>> remove(243132, 2)
                                                   Implement without the template, then change
    4313
                                                   your implementation to match the template.
    .....
                                                   OR
    kept, digits = 0, 0
                                                   If the template is helpful, use it.
    while n > 0
                                                  Annotate names with values from your chosen
                                                   example
        n, last = n // 10, n % 10
                last != digit
                                                  Write code to compute the result
        if
                       kept
                                    last
                                                  Did you really return the right thing?
            kept =
      21
            digits = _____
                                                  Check your solution with the other examples
                   kept
    return
```

```
def remove(n, digit):
    """"Return all'digits of non-negative N
                                                  Read the description
                       IT, for some
                   3
             ire
       231
                                                  Verify the examples & pick a simple one
                       IT less than 10.
             eaa
    >>> remove(231, 3)
                                                  Read the template
    21
    >>> remove(243132, 2)
                                                  Implement without the template, then change
    4313
                                                  your implementation to match the template.
    .....
                                                  OR
    kept, digits = 0, 0
                                                  If the template is helpful, use it.
    while n > 0
                                                  Annotate names with values from your chosen
                                                  example
        n, last = n // 10, n % 10
                last != digit
                                                  Write code to compute the result
        if
                       kept/10 +
                                    last
                                                  Did you really return the right thing?
            kept =
      21
            digits = _____
                                                  Check your solution with the other examples
                   kept
    return
```

```
def remove(n, digit):
    """"Return all'digits of non-negative N
                                                  Read the description
                       IT, for some
                   3
             ire
       231
                                                  Verify the examples & pick a simple one
                       IT less than 10.
             eaa
    >>> remove(231, 3)
                                                  Read the template
    21
    >>> remove(243132, 2)
                                                  Implement without the template, then change
    4313
                                                  your implementation to match the template.
    .....
                                                  OR
    kept, digits = 0, 0
                                                  If the template is helpful, use it.
    while n > 0
                                                  Annotate names with values from your chosen
                                                  example
        n, last = n // 10, n % 10
                last != digit
                                                  Write code to compute the result
        if
                       kept/10 +
                                    last
                                                  Did you really return the right thing?
            kept =
      21
            digits = _____
                                                  Check your solution with the other examples
                   kept * 10
    return
```

```
def remove(n, digit):
    ali diaits of non-negative N
                                                 Read the description
                      IT, for some
                  3
            ire
       231
                                                 Verify the examples & pick a simple one
                      IT less than 10.
             eaa
   >>> remove(231, 3)
                                                 Read the template
    21
   >>> remove(243132, 2)
                                                 Implement without the template, then change
   4313
                                                 your implementation to match the template.
    .....
                                                 OR
    kept, digits = 0, 0
                                                 If the template is helpful, use it.
                   n > 0
   while
                                                 Annotate names with values from your chosen
                                                 example
       n, last = n // 10, n % 10
                last != digit
                                                 Write code to compute the result
        if
                      kept/10 +
                                  last
                                                 Did you really return the right thing?
            kept =
                     digits + 1
      21
            digits =
                                                 Check your solution with the other examples
                  kept * 10
    return
```

```
def remove(n, digit):
    """"Return all'digits of non-negative N
                                                   Read the description
                       IT, for some
             ire
                   3
       231
                        IT less than 10.
                                                   Verify the examples & pick a simple one
             eaa
    >>> remove(231, 3)
                                                   Read the template
    21
    >>> remove(243132, 2)
                                                   Implement without the template, then change
    4313
                                                   your implementation to match the template.
    .....
                                                   OR
    kept, digits = 0, 0
                                                   If the template is helpful, use it.
    while n > 0
                                                   Annotate names with values from your chosen
                                                   example
        n, last = n // 10, n % 10
                last != digit
                                                   Write code to compute the result
        if
                        kept/10 +
                                    last
                                                   Did you really return the right thing?
            kept =
                       digits + 1
      21
            digits =
                                                   Check your solution with the other examples
                   kept * 10 ** (digits-1)
    return
```

```
def remove(n, digit):
    """"Return all'digits of non-negative N
                                                   Read the description
                       IT, for some
                   3
             ire
       231
                        IT less than 10.
                                                   Verify the examples & pick a simple one
             eaa
    >>> remove(231, 3)
                                                   Read the template
    21
    >>> remove(243132, 2)
                                                   Implement without the template, then change
    4313
                                                   your implementation to match the template.
    .....
                                                   OR
    kept, digits = 0, 0
                                                   If the template is helpful, use it.
    while n > 0
                                                   Annotate names with values from your chosen
                                                   example
        n, last = n // 10, n % 10
                last != digit
                                                   Write code to compute the result
        if
                       kept/10 +
                                    last
                                                   Did you really return the right thing?
            kept =
                      digits + 1
      21
            digits =
                                                   Check your solution with the other examples
             round(kept * 10 ** (digits-1))
    return
```