NESTED LOOPS

(Supplementary Notes)

Question 4 -- From Lab 8 (Least Common Multiple)

Write a function **`smallest_multiple`** that takes as input a *positive integer* **`n`** and returns the smallest number **`x`**, such that **`x % j == 0`**, for all **`1 <= j <= n`**.

For example,
for input n=5, the function should return 60

as it is the smallest number that can be divided by each of the numbers from 1 to 5, without any remainder.

Question 4 -- From Lab 8 (Least Common Multiple)

Write a function **`smallest_multiple`** that takes as input a *positive integer* **`n`** and returns the smallest number **`x`**, such that **`x % j == 0`**, for all **`1 <= j <= n`**.

For example,
for input n=5, the function should return 60

as it is the smallest number that can be divided by each of the numbers from 1 to 5 without any remainder.

x		1<=j<=n		
60	%	1	== 0	✅
60	%	2	== 0	✅
60	%	3	== 0	✅
60	%	4	== 0	✅
60	%	5	== 0	✅

How do you find this magical number x?

You don’t know how big or how small this magical number is...

In search of this magical number, we’ll just have to traverse the entire number line, checking our condition(s) on each number


```
i     1<=j<=n
i  %   1     == 0  ?
i  %   2     == 0  ?
i  %   …     == 0  ?
i  %   …     == 0  ?
i  %   n     == 0  ?
```

Smallest Multiple

i	OUTER LOOP					
	1					
	2					
	3					
	4					
	....					
	x					
	....					
	+ infinity					

i		1<=j<=n			
i	%	1	==	0	?
i	%	2	==	0	?
i	%	3	==	0	?
i	%	…	==	0	?
i	%	n	==	0	?

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j →			
↓	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
i     1<=j<=n
i  %    1     == 0  ?
i  %    2     == 0  ?
i  %    3     == 0  ?
i  %    ...   == 0  ?
i  %    n     == 0  ?
```

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j →			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
↓	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
     return result
```

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1                          initialization
    result = None
    while result == None:          The stopping condition is NOT dependent on i
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1                  update
    return result
```

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):

    i = 1

    result = None

    while result == None:

        j = 1

        while j <= n:

            if i % j == 0:

                j = j + 1

            else:

                break

        if j == n+1:

            result = i

        i = i + 1

    return result
```

initialization

stopping condition

update

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Check condition

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

TRUE

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Update j

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Check condition for new value of j

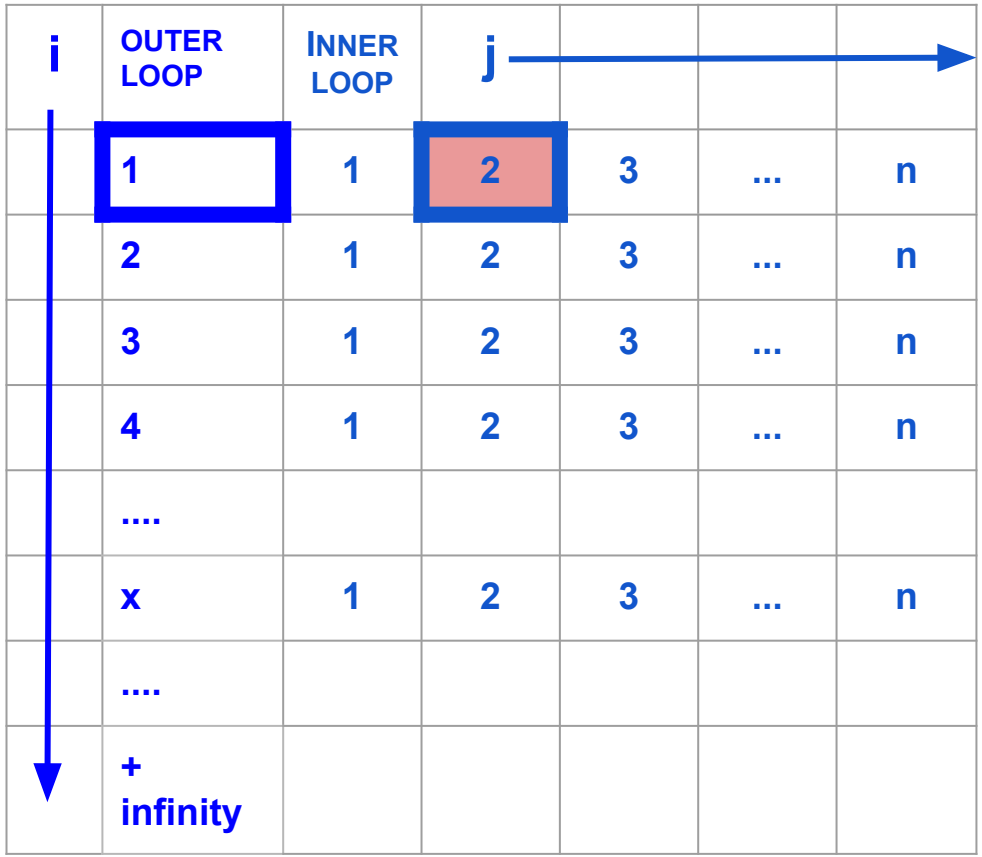
Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
     return result
```

FALSE

Smallest Multiple




```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

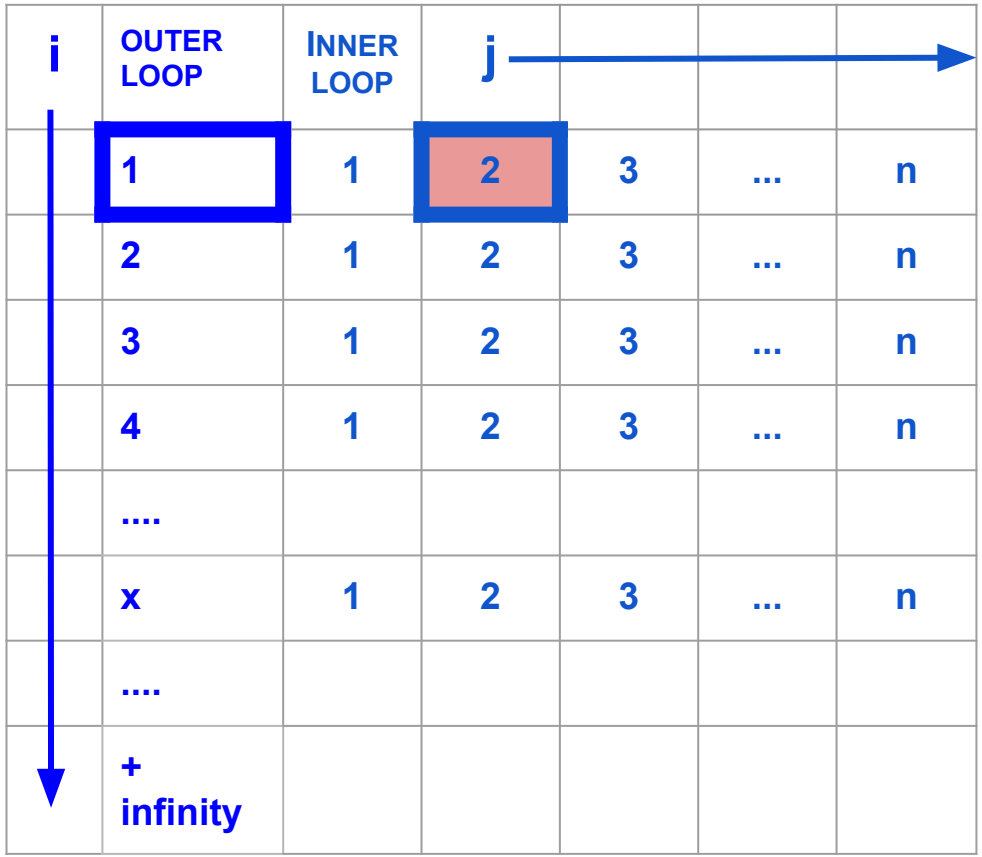
For this i, no need to check for further j

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Smallest Multiple




```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Move onto next i

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Reinitialize j with 1. Starting over with new i

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Check condition

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

TRUE, check for next j

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

TRUE for next j

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

FALSE

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Eventually.. You'll get to the number you are looking for

x % 1 == 0 ✅

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

x % 1 == 0 ✅

x % 2 == 0 ✅

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

x % 1 == 0 ✓
x % 2 == 0 ✓
x % 3 == 0 ✓

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

x % 1 == 0 ✓
x % 2 == 0 ✓
x % 3 == 0 ✓
x % : == 0 ✓
x % n == 0 ✓

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

j is updated to n + 1

x % 1 == 0 ✓
x % 2 == 0 ✓
x % 3 == 0 ✓
x % : == 0 ✓
x % n == 0 ✓

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

n + 1 > n

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:          <- True
            result = i
        i = i + 1
    return result
```

Smallest Multiple

i	OUTER LOOP	INNER LOOP	j			
	1	1	2	3	...	n
	2	1	2	3	...	n
	3	1	2	3	...	n
	4	1	2	3	...	n
	....					
	x	1	2	3	...	n
	....					
	+ infinity					

```
def smallest_multiple(n=5):
    i = 1
    result = None
    while result == None:
        j = 1
        while j <= n:
            if i % j == 0:
                j = j + 1
            else:
                break
        if j == n+1:
            result = i
        i = i + 1
    return result
```

Largest Palindrome Problem

Largest Palindrome Problem (n=3)

i	OUTER LOOP	INNER LOOP	j →			
	100	100	101	102	...	999
	101	100	101	102	...	999
	102	100	101	102	...	999
	103	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	999	100	101	102	...	999

Find the largest palindrome obtained from multiplying two n-digit numbers

For each value of (`i, j`) pair

1. Check if **`i*j`** is a palindrome
2. Check if **`i*j`** is the **largest palindrome** you've seen so far

Largest Palindrome Problem (n=3)

i	OUTER LOOP	INNER LOOP	j →			
	100	100	101	102	...	999
	101	100	101	102	...	999
	102	100	101	102	...	999
	103	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	999	100	101	102	...	999

Find the largest palindrome obtained from multiplying two n-digit numbers

For each value of (`i, j`) pair

1. Check if **`i*j`** is a palindrome
2. Check if **`i*j`** is the **largest palindrome** you've seen so far

Largest Palindrome Problem (n=3)

i	OUTER LOOP	INNER LOOP	j →			
	100	100	101	102	...	999
	101	100	101	102	...	999
	102	100	101	102	...	999
	103	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
↓	999	100	101	102	...	999

Find the largest palindrome obtained from multiplying two n-digit numbers

For each value of (`i, j`) pair

1. Check if **`i*j`** is a palindrome
2. Check if **`i*j`** is the **largest palindrome** you've seen so far

Largest Palindrome Problem (n=3)

i	OUTER LOOP	INNER LOOP	j →			
	100	100	101	102	...	999
	101	100	101	102	...	999
	102	100	101	102	...	999
	103	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
↓	999	100	101	102	...	999

Find the largest palindrome obtained from multiplying two n-digit numbers

For each value of (`i, j`) pair

1. Check if **`i*j`** is a palindrome
2. Check if **`i*j`** is the **largest palindrome** you've seen so far

Largest Palindrome Problem (n=3)

i	OUTER LOOP	INNER LOOP	j →			
	100	100	101	102	...	999
	101	100	101	102	...	999
	102	100	101	102	...	999
	103	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	999	100	101	102	...	999

Find the largest palindrome obtained from multiplying two n-digit numbers

For each value of (`i, j`) pair

1. Check if **`i*j`** is a palindrome
2. Check if **`i*j`** is the **largest palindrome** you've seen so far

Largest Palindrome Problem (n=3)

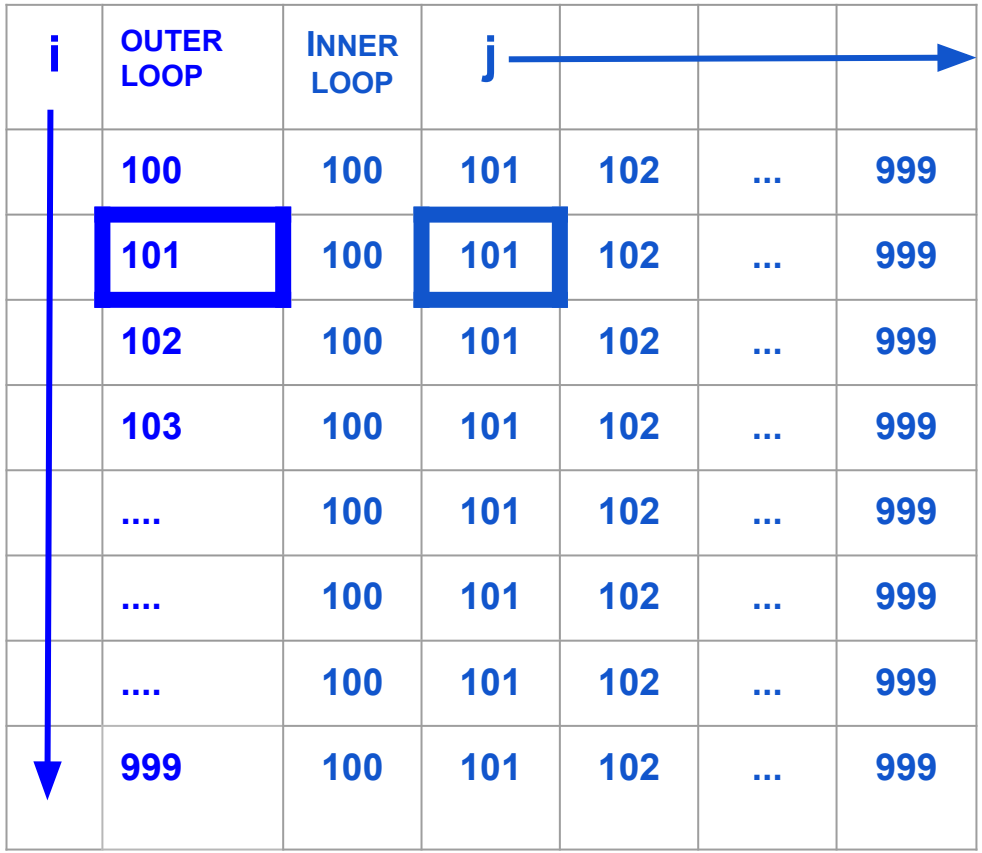
i	OUTER LOOP	INNER LOOP	j →			
	100	100	101	102	...	999
	101	100	101	102	...	999
	102	100	101	102	...	999
	103	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	999	100	101	102	...	999

Find the largest palindrome obtained from multiplying two n-digit numbers

For each value of (`i, j`) pair

1. Check if **`i*j`** is a palindrome
2. Check if **`i*j`** is the **largest palindrome** you've seen so far

Largest Palindrome Problem (n=3)



Find the largest palindrome obtained from multiplying two n-digit numbers

For each value of (`i, j`) pair

1. Check if **`i*j`** is a palindrome
2. Check if **`i*j`** is the **largest palindrome** you've seen so far

Largest Palindrome Problem (n=3)

i	OUTER LOOP	INNER LOOP	j			
	100	100	101	102	...	999
	101	100	101	102	...	999
	102	100	101	102	...	999
	103	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	999	100	101	102	...	999

Find the largest palindrome obtained from multiplying two n-digit numbers

For each value of (`i, j`) pair

1. Check if **`i*j`** is a palindrome
2. Check if **`i*j`** is the **largest palindrome** you've seen so far

Largest Palindrome Problem (n=3)

i	OUTER LOOP	INNER LOOP	j →			
	100	100	101	102	...	999
	101	100	101	102	...	999
	102	100	101	102	...	999
	103	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	999	100	101	102	...	999

Find the largest palindrome obtained from multiplying two n-digit numbers

For each value of (`i, j`) pair

1. Check if **`i*j`** is a palindrome
2. Check if **`i*j`** is the **largest palindrome** you've seen so far

Largest Palindrome Problem (n=3)

i	OUTER LOOP	INNER LOOP	j →			
	100	100	101	102	...	999
	101	100	101	102	...	999
	102	100	101	102	...	999
	103	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
	....	100	101	102	...	999
↓	999	100	101	102	...	999

Find the largest palindrome obtained from multiplying two n-digit numbers

For each value of (`i, j`) pair

1. Check if **`i*j`** is a palindrome
2. Check if **`i*j`** is the **largest palindrome** you've seen so far

Assignment Questions

Question 1.

For `n=4`, the function should return `1634` since it is the smallest number that can be written as the sum of fourth power of its digits i.e.

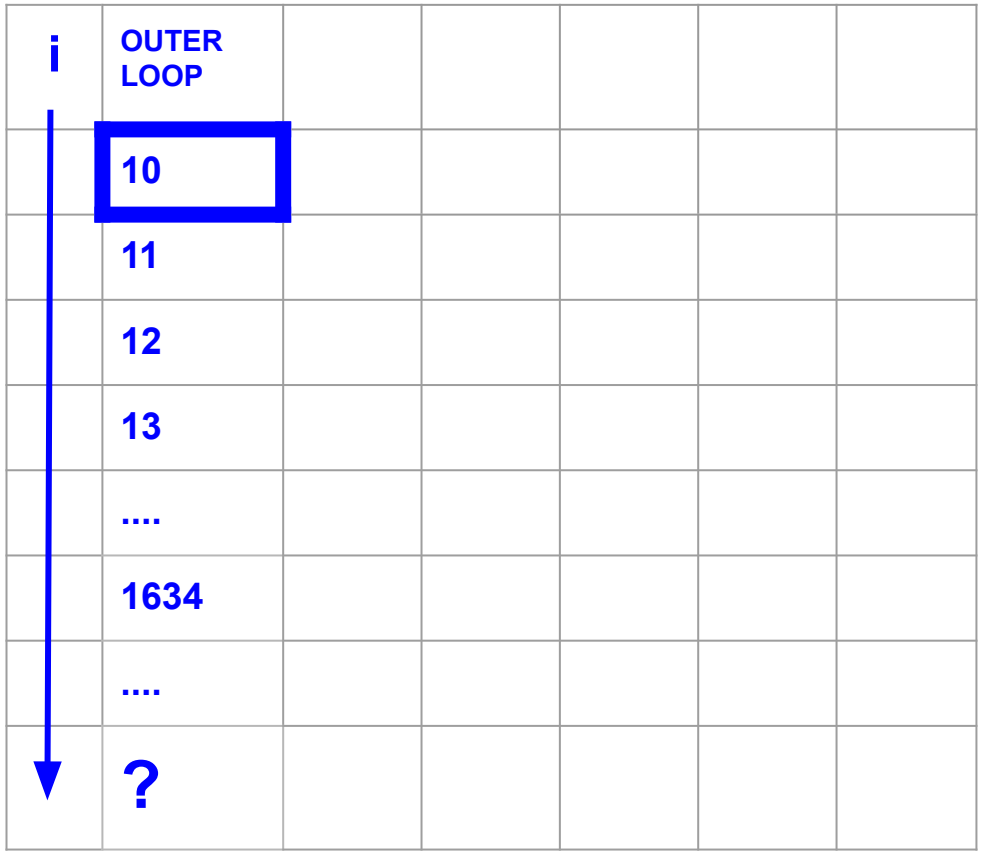
$$1634 = 1^4 + 6^4 + 3^4 + 4^4$$

As $1 = 1^4$ is not a sum, it is not included.

Similarly, for input `n=5`, the function should output `4150` since:

$$4150 = 4^5 + 1^5 + 5^5 + 0^5$$

Question 1. (n=4)



n=4

$1634 = 1^4 + 6^4 + 3^4 + 4^4$

As $1 = 1^4$ is not a sum, it is not included.

Question 1. (n=4)

i	OUTER LOOP	INNER LOOP	j			
	10	?	?			
	11	?	?			
	12	?	?			
	13	?	?			
	....					
	1634	?	?	?	?	
	....					
	+∞					

n=4

$1634 = 1^4 + 6^4 + 3^4 + 4^4$

As $1 = 1^4$ is not a sum, it is not included.

Question 1. (n=4)

i	OUTER LOOP	INNER LOOP	j →			
	10	1^4	0^4			
	11	1^4	1^4			
	12	1^4	2^4			
	13	1^4	3^4			
	....					
	1634	1^4	6^4	3^4	4^4	
	....					
↓	$+\infty$					

n=4

$1634 = 1^4 + 6^4 + 3^4 + 4^4$

As $1 = 1^4$ is not a sum, it is not included.

- Remember: `i % 10` and `i // 10` are your friends
- Keep a running sum of digits raised to the power of n

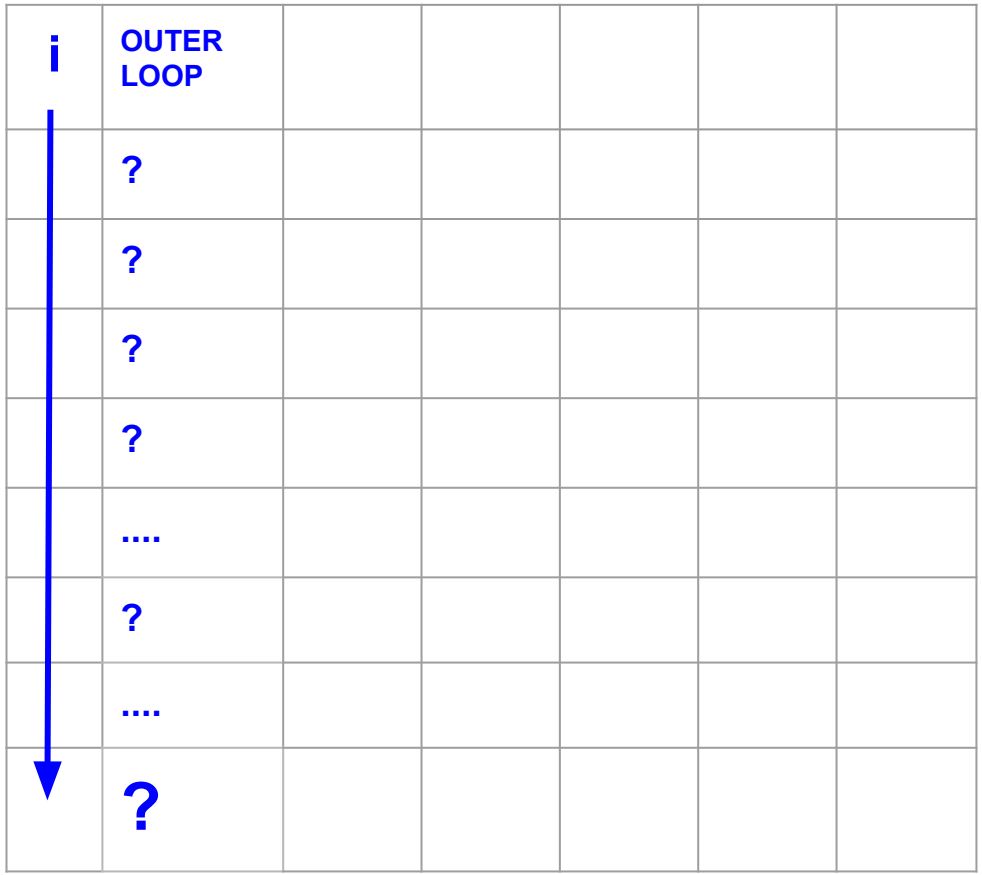
Question 2. Goldbach's Conjecture:

Every even natural number greater than 2 is the sum of two prime numbers.

Given an even natural number greater than 2, find the smallest two primes that sum to the given natural number

Return the product of these two prime numbers

Question 2. Goldbach's Conjecture (n=10)



Find the smallest two primes that sum to the given natural even number greater than 2

Question 2. Goldbach's Conjecture (n=10)

i	OUTER LOOP	INNER LOOP	j			
	1	?	?	?	....	?
	2	?	?	?	....	?
	3	?	?	?	....	?
	5	?	?	?	....	?
	7	?	?	?	....	?
	11	?	?	?	....	?
	....					
	$+\infty$					

Find the smallest two primes that sum to the given natural even number greater than 2

Question 2. Goldbach's Conjecture (n=10)

i	OUTER LOOP	INNER LOOP	j	→			
	1	1	2	3	5	....	+∞
	2	1	2	3	5	....	+∞
	3	1	2	3	5	....	+∞
	5	1	2	3	5	....	+∞
	7	1	2	3	5	....	+∞
	11	1	2	3	5	....	+∞
	....						
	+∞						

Find the smallest two primes that sum to the given natural even number greater than 2

For every pair of prime numbers (i, j)

Check if they meet the requirements

Question 2. Goldbach's Conjecture (n=10)

i	OUTER LOOP	INNER LOOP	j				
	1	1	2	3	5	....	+∞
	2	1	2	3	5	....	+∞
	3	1	2	3	5	7	....
	5	1	2	3	5	....	+∞
	7	1	2	3	5	....	+∞
	11	1	2	3	5	....	+∞
	....						
	+∞						

Find the smallest two primes that sum to the given natural even number greater than 2

For every pair of prime numbers (i, j)

Check if they meet the requirements

Question 3. Goldbach's (disproven) Conjecture:

Every odd composite number can be written as sum of a prime number and twice a square.

$c = p + 2x^2$

c: composition number
p: prime number
x: an arbitrary number

Question 3. Goldbach's Conjecture (n=9)

i	OUTER LOOP	INNER LOOP	j				
	?	?	?	?	?	....	?
	?	?	?	?	?	....	?
	?	?	?	?	?	....	?
	?	?	?	?	?	....	?
	....	?	?	?	?	....	?
	?	?	?	?	?	....	?
	....						
	?						

Every odd composite number can be written as sum of a prime number and twice a square.

$c = p + 2x^2$

c: odd composite number
p: prime number
x: an arbitrary number

Example:

n= 9

$9 = \mathbf{7} + 2\times\mathbf{1}^2$