

Exercise 3.1

Write a script that reads the current time and converts it to a time of day in hours, minutes, and seconds, plus the number of days since the epoch.

```
In [2]: # Answer for Exercise 3.1

import time

# Get current time in seconds since epoch
current_time = time.time()

# Calculate days since epoch
days_since_epoch = int(current_time // (24 * 3600))

# Calculate remaining seconds
seconds_today = current_time % (24 * 3600)

# Convert remaining seconds to hours, minutes, and seconds
hours = int(seconds_today // 3600)
minutes = int((seconds_today % 3600) // 60)
seconds = int(seconds_today % 60)

# Print the results
print(f"Current time: {hours:02d}:{minutes:02d}:{seconds:02d}")
print(f"Days since epoch: {days_since_epoch}")
```

Current time: 17:21:48
Days since epoch: 19956

Exercise 3.2

Question 1

Write a function named `check_fermat` that takes four parameters—`a`, `b`, `c` and `n`—and checks to see if Fermat's theorem holds. If `n` is greater than 2 and $a^n + b^n = c^n$, the program should print, "Fermat was wrong!" Otherwise the program should print, "No, that doesn't work."

```
In [3]: # Answer for Question 1

def check_fermat(a, b, c, n):
    if n > 2 and (a**n + b**n == c**n):
        print("Fermat was wrong!")
    else:
        print("No, that doesn't work.")

# Test the function
check_fermat(3, 4, 5, 2) # Should print "No, that doesn't work."
check_fermat(3, 4, 5, 3) # Should print "No, that doesn't work."
```

No, that doesn't work.
No, that doesn't work.

Question 2

Write a function that prompts the user to input values for a, b, c and n, converts them to integers, and uses `check_fermat` to check whether they violate Fermat's theorem.

```
In [4]: # Answer for Question 2

def fermt_check():
    a = int(input("Enter a value for a: "))
    b = int(input("Enter a value for b: "))
    c = int(input("Enter a value for c: "))
    n = int(input("Enter a value for n: "))

    check_fermat(a, b, c, n)

# Run the function
fermt_check()
```

No, that doesn't work.

Exercise 3.3

Question 1

Write a function named `is_triangle` that takes three integers as arguments, and that prints either "Yes" or "No", depending on whether you can or cannot form a triangle from sticks with the given lengths.

```
In [5]: # Answer for Question 1

def is_triangle(a, b, c):
    if a + b > c and b + c > a and c + a > b:
        print("Yes")
    else:
        print("No")

# Test the function
is_triangle(3, 4, 5) # Should print "Yes"
is_triangle(1, 1, 12) # Should print "No"
```

Yes

No

Question 2

Write a function that prompts the user to input three stick lengths, converts them to integers, and uses `is_triangle` to check whether sticks with the given lengths can form a triangle.

```
In [6]: # Answer for Question 2
```

```
def check_triangle():
    try:
```

```
a = int(input("Enter the length of the first stick: "))
b = int(input("Enter the length of the second stick: "))
c = int(input("Enter the length of the third stick: "))

if a <= 0 or b <= 0 or c <= 0:
    print("Stick lengths must be positive integers.")
else:
    is_triangle(a, b, c)
except ValueError:
    print("Invalid input. Please enter integer values.")

# Run the function
check_triangle()
```

Yes