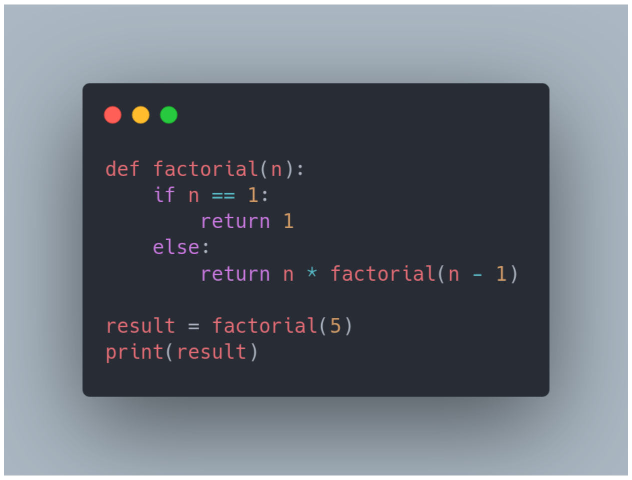
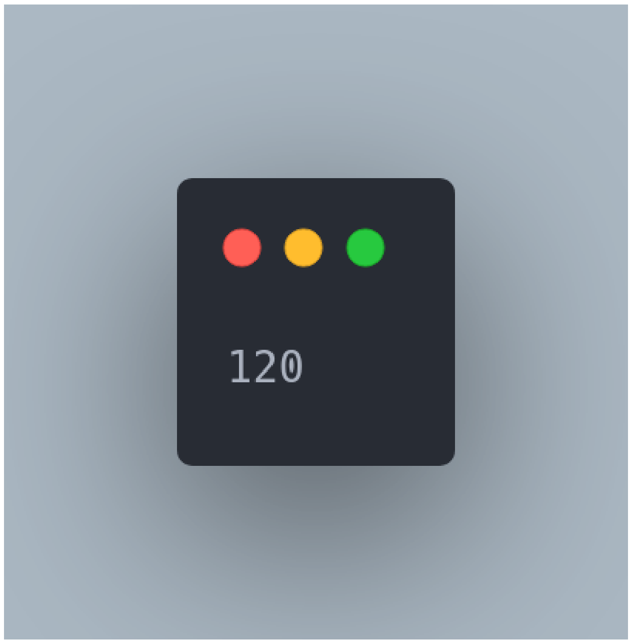
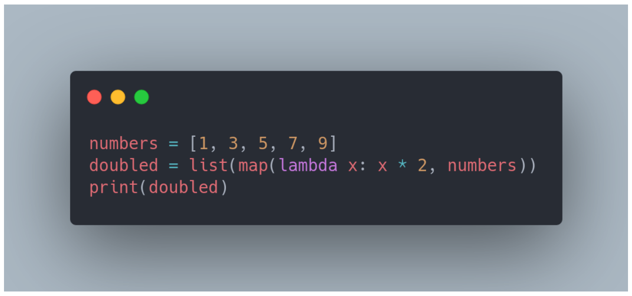


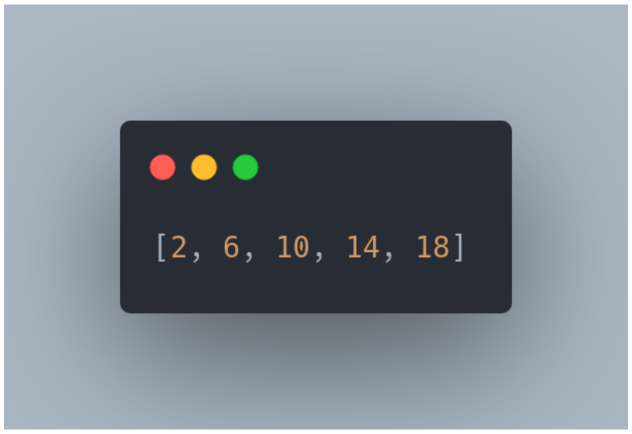
```python
def factorial(n):
    if n == 1:
        return 1
    else:
        return n * factorial(n - 1)

result = factorial(5)
print(result)
```



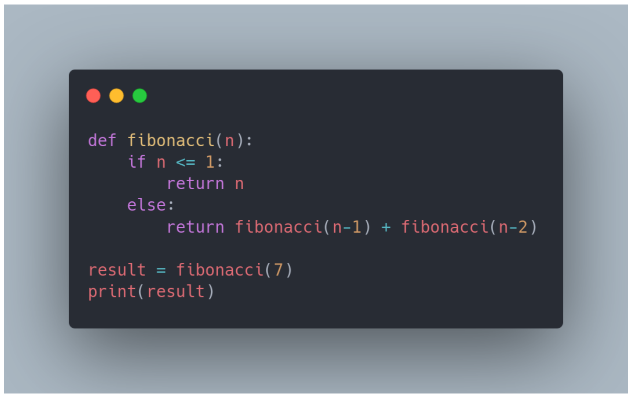
```
120
```



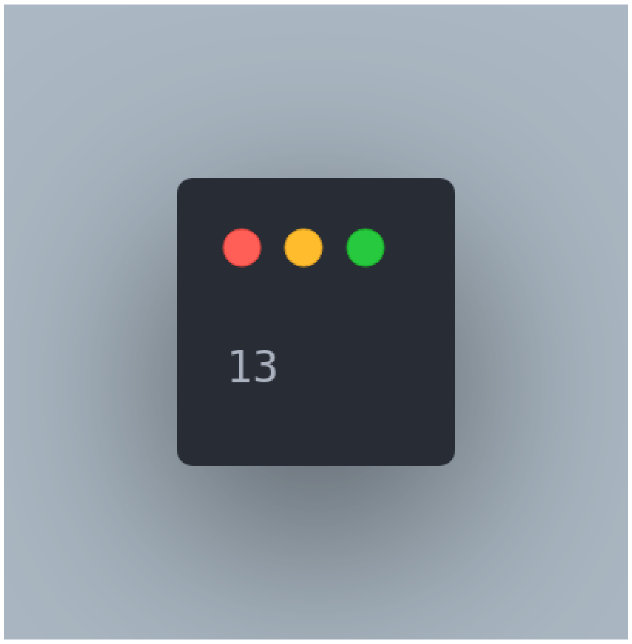
```python
numbers = [1, 3, 5, 7, 9]
doubled = list(map(lambda x: x * 2, numbers))
print(doubled)
```



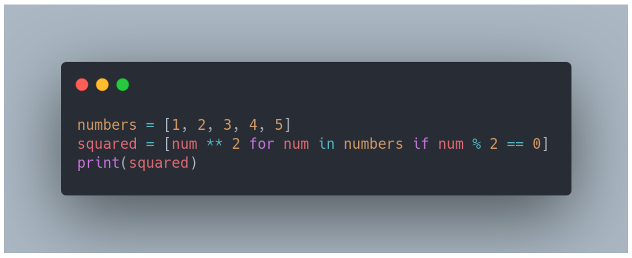
```
[2, 6, 10, 14, 18]
```



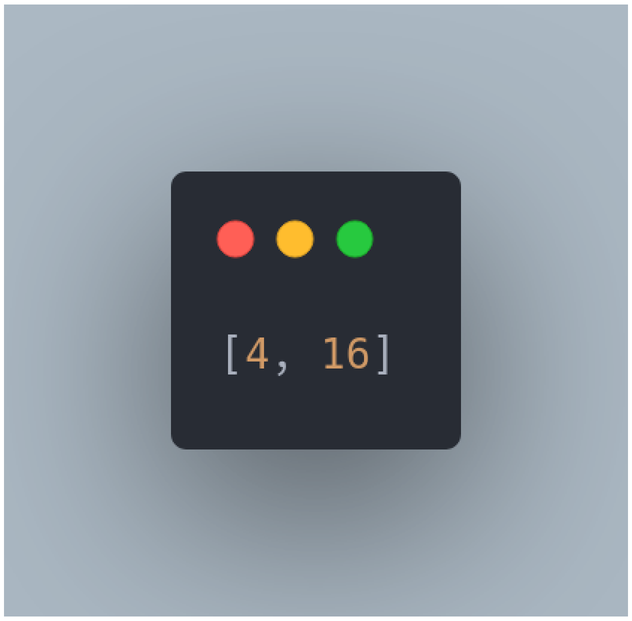
```python
def fibonacci(n):
    if n <= 1:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)

result = fibonacci(7)
print(result)
```



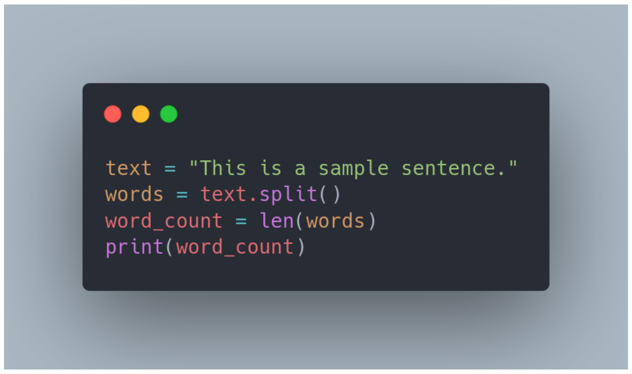
```
13
```



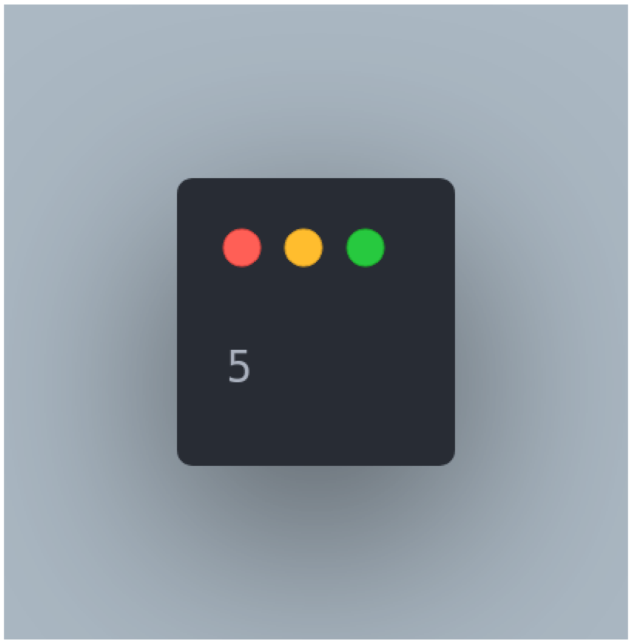
```python
numbers = [1, 2, 3, 4, 5]
squared = [num ** 2 for num in numbers if num % 2 == 0]
print(squared)
```



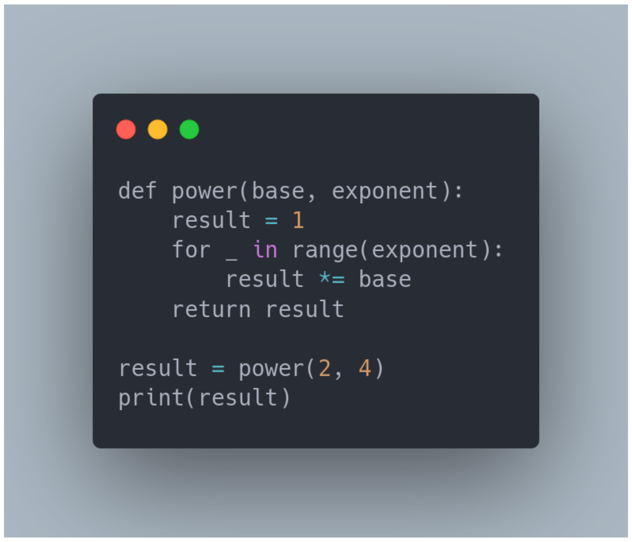
```
[4, 16]
```



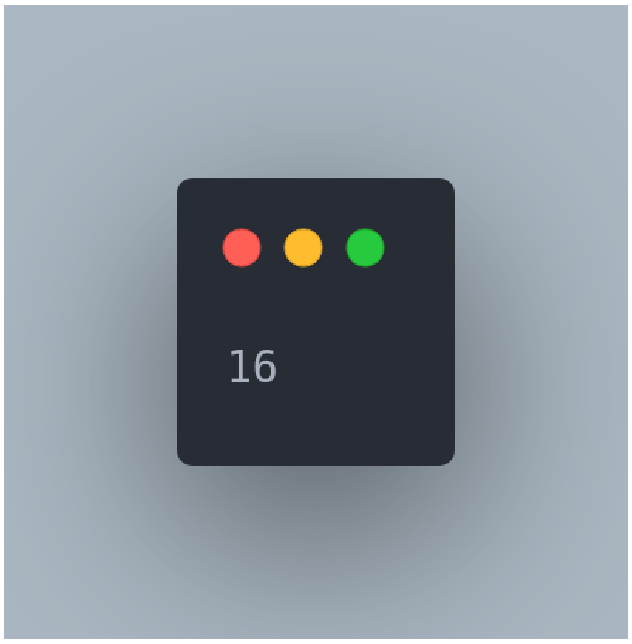
```python
text = "This is a sample sentence."
words = text.split()
word_count = len(words)
print(word_count)
```



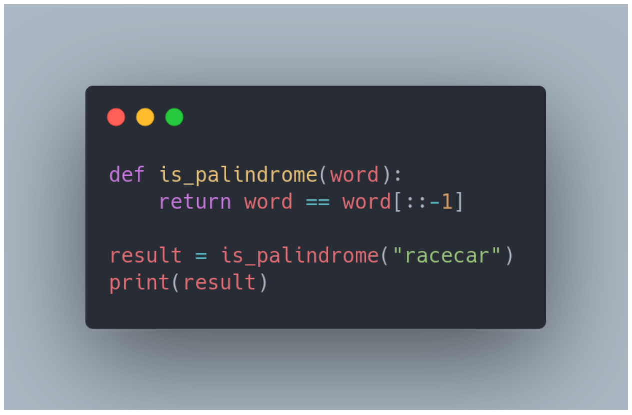
```
5
```



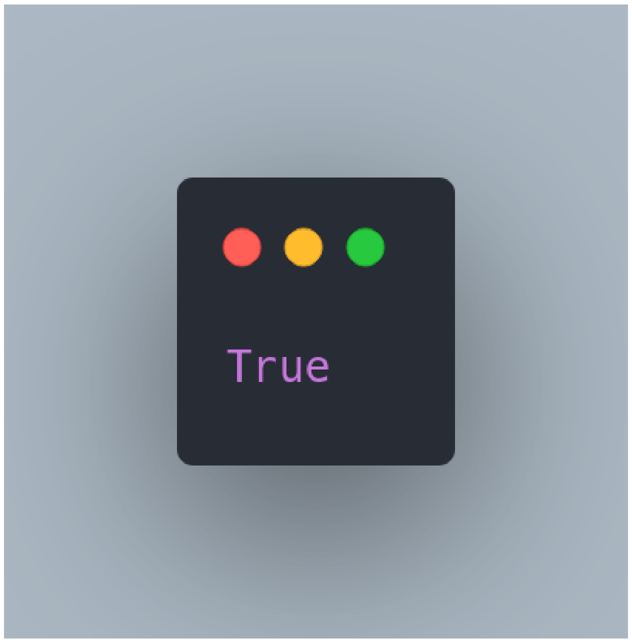
```python
def power(base, exponent):
    result = 1
    for _ in range(exponent):
        result *= base
    return result

result = power(2, 4)
print(result)
```



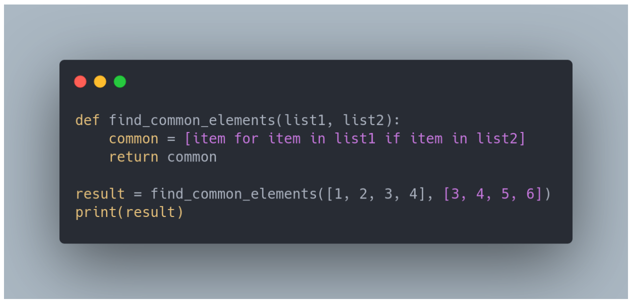
```
16
```



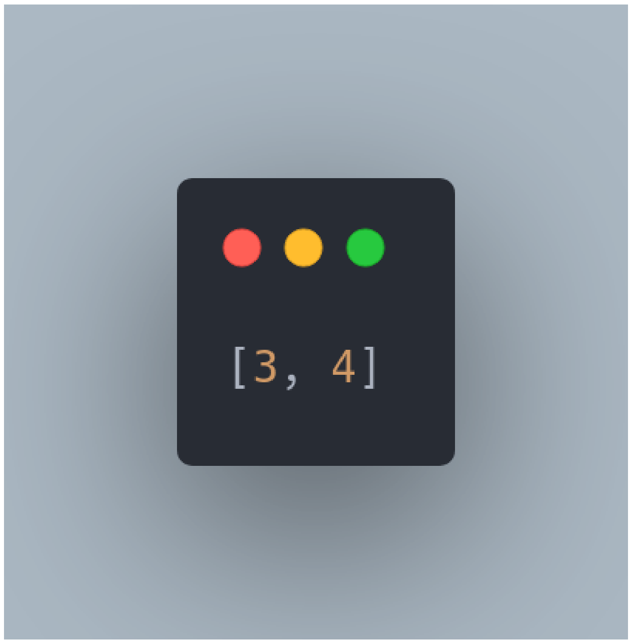
```python
def is_palindrome(word):
    return word == word[::-1]

result = is_palindrome("racecar")
print(result)
```



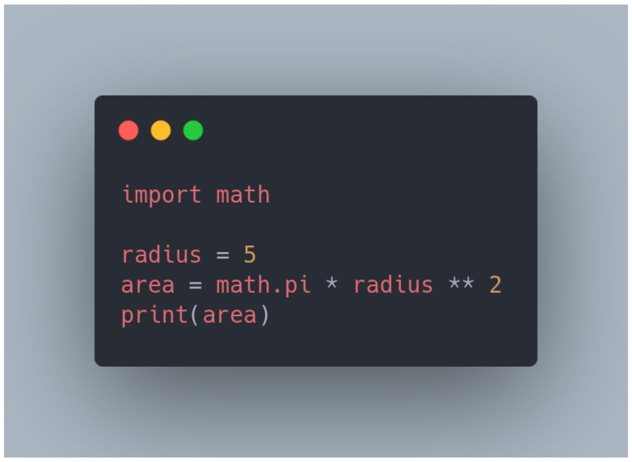
```
True
```



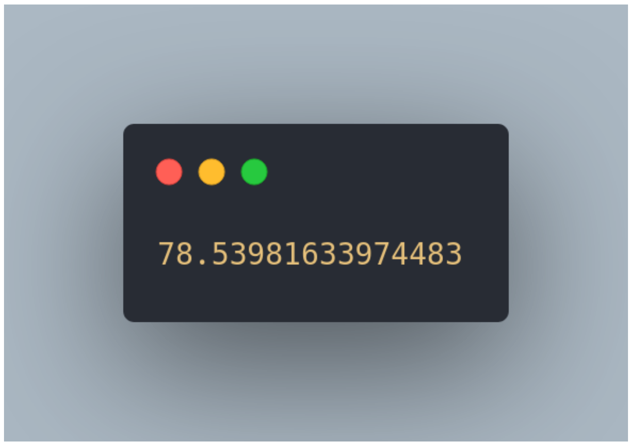
```python
def find_common_elements(list1, list2):
    common = [item for item in list1 if item in list2]
    return common

result = find_common_elements([1, 2, 3, 4], [3, 4, 5, 6])
print(result)
```



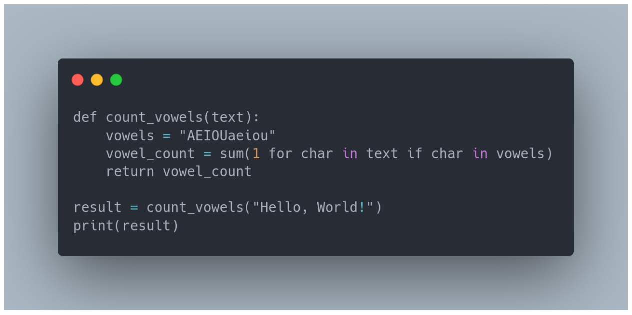
```
[3, 4]
```



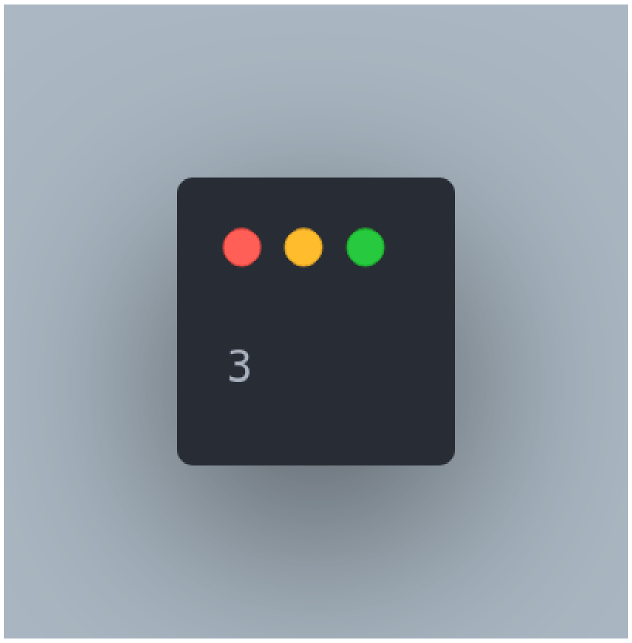
```python
import math

radius = 5
area = math.pi * radius ** 2
print(area)
```



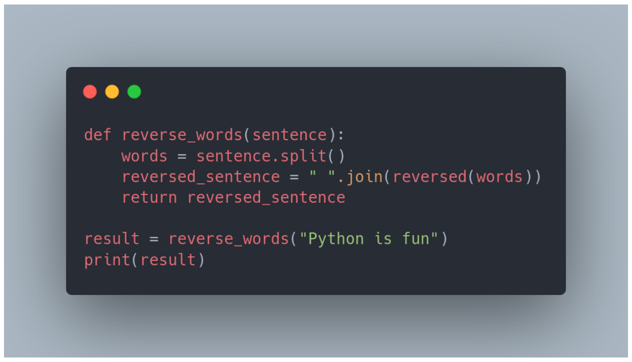
```
78.53981633974483
```



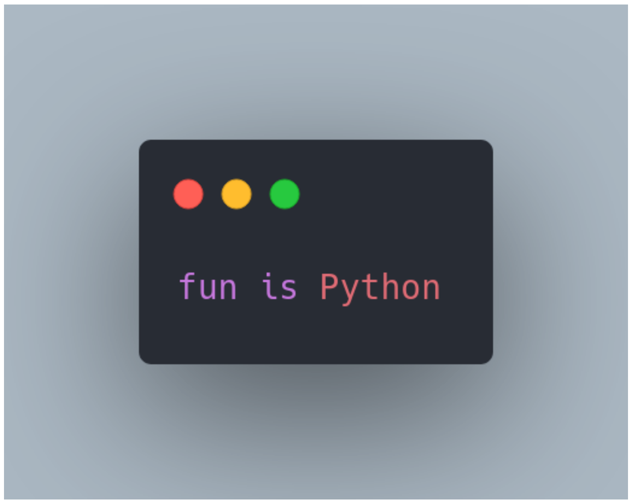
```python
def count_vowels(text):
    vowels = "AEIOUaeiou"
    vowel_count = sum(1 for char in text if char in vowels)
    return vowel_count

result = count_vowels("Hello, World!")
print(result)
```



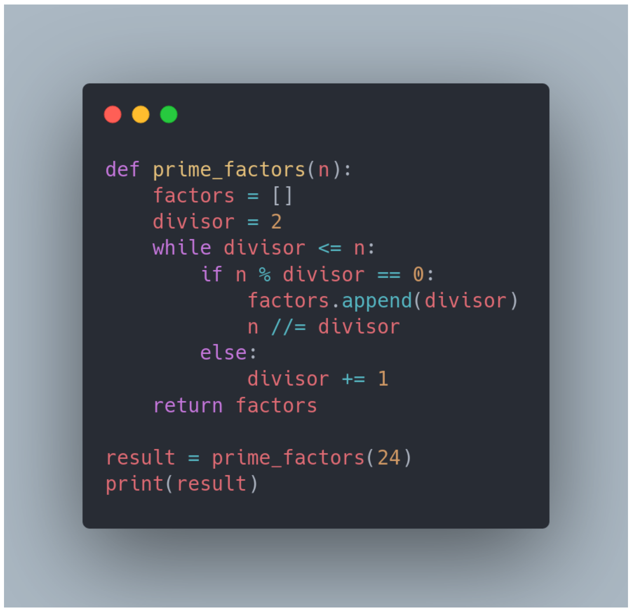
```
3
```



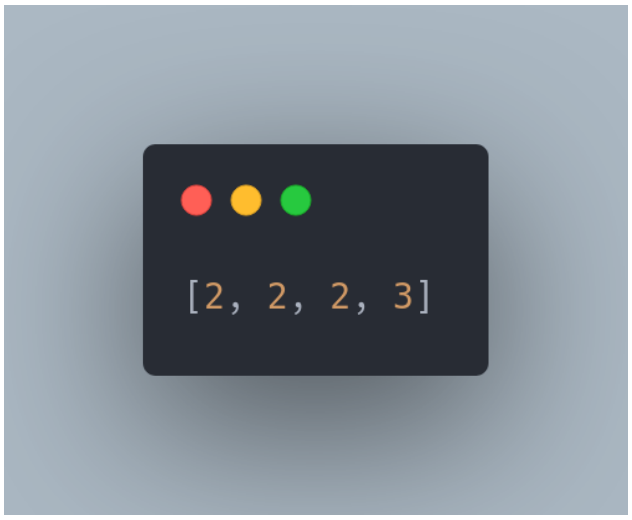
```python
def reverse_words(sentence):
    words = sentence.split()
    reversed_sentence = " ".join(reversed(words))
    return reversed_sentence

result = reverse_words("Python is fun")
print(result)
```



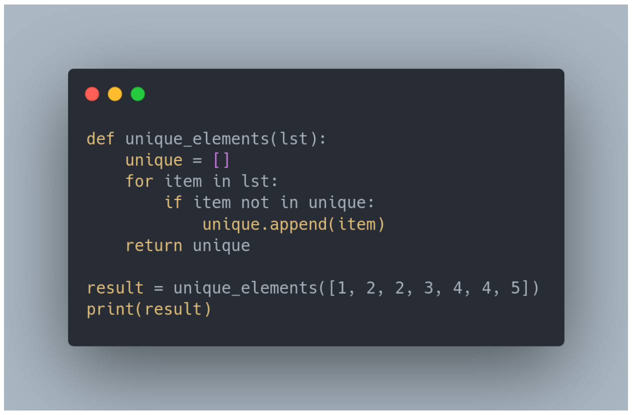
```
fun is Python
```



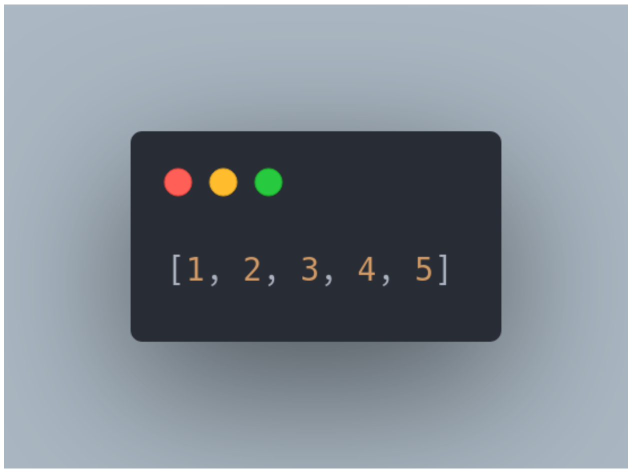
```python
def prime_factors(n):
    factors = []
    divisor = 2
    while divisor <= n:
        if n % divisor == 0:
            factors.append(divisor)
            n //= divisor
        else:
            divisor += 1
    return factors

result = prime_factors(24)
print(result)
```



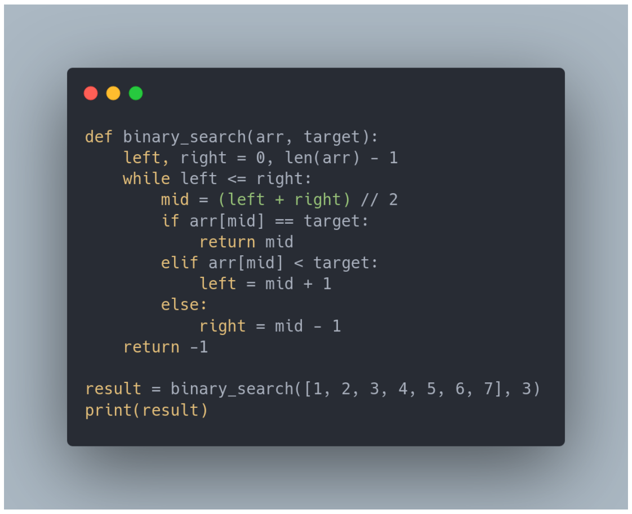
```
[2, 2, 2, 3]
```



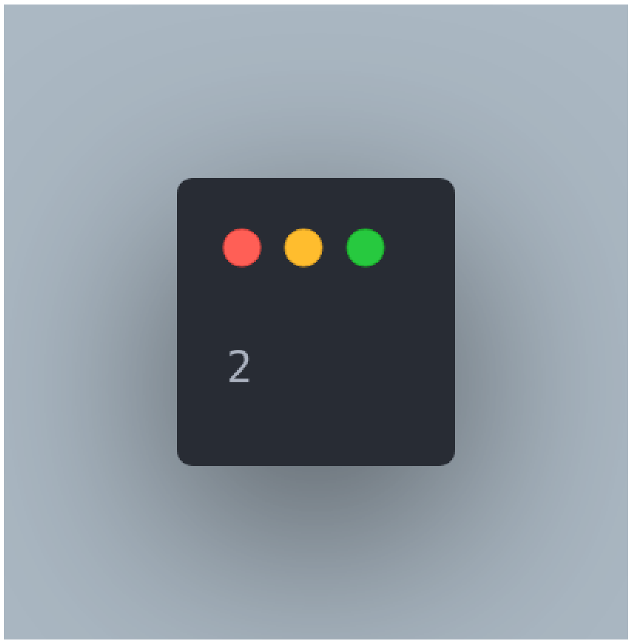
```python
def unique_elements(lst):
    unique = []
    for item in lst:
        if item not in unique:
            unique.append(item)
    return unique

result = unique_elements([1, 2, 2, 3, 4, 4, 5])
print(result)
```



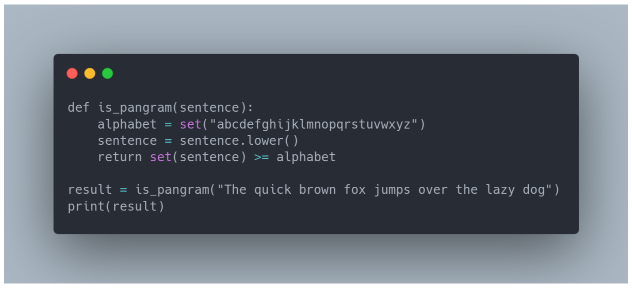
```
[1, 2, 3, 4, 5]
```



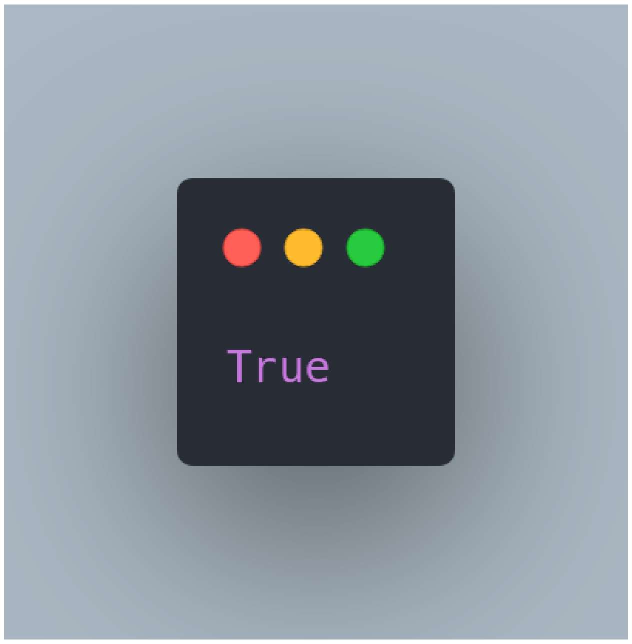
```python
def binary_search(arr, target):
    left, right = 0, len(arr) - 1
    while left <= right:
        mid = (left + right) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return -1

result = binary_search([1, 2, 3, 4, 5, 6, 7], 3)
print(result)
```



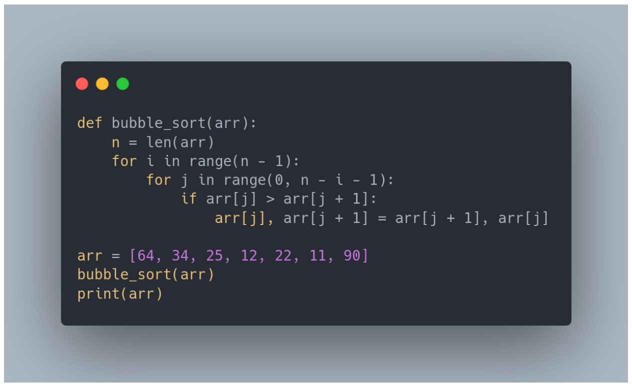
```
2
```



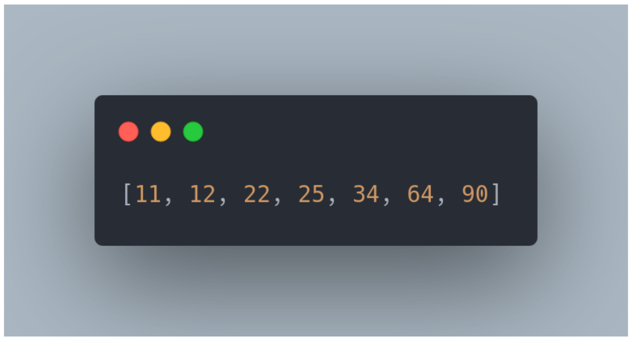
```python
def is_pangram(sentence):
    alphabet = set("abcdefghijklmnopqrstuvwxyz")
    sentence = sentence.lower()
    return set(sentence) >= alphabet

result = is_pangram("The quick brown fox jumps over the lazy dog")
print(result)
```



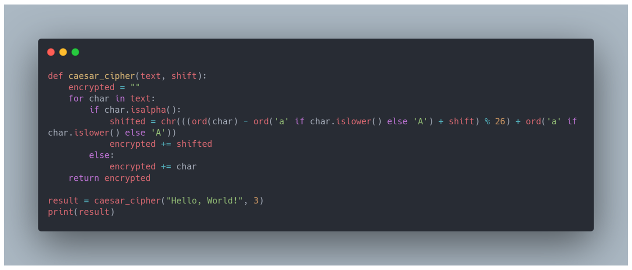
```
True
```



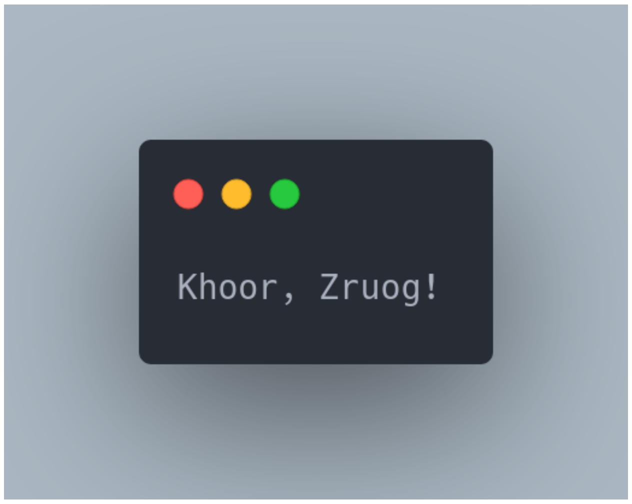
```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n - 1):
        for j in range(0, n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]

arr = [64, 34, 25, 12, 22, 11, 90]
bubble_sort(arr)
print(arr)
```



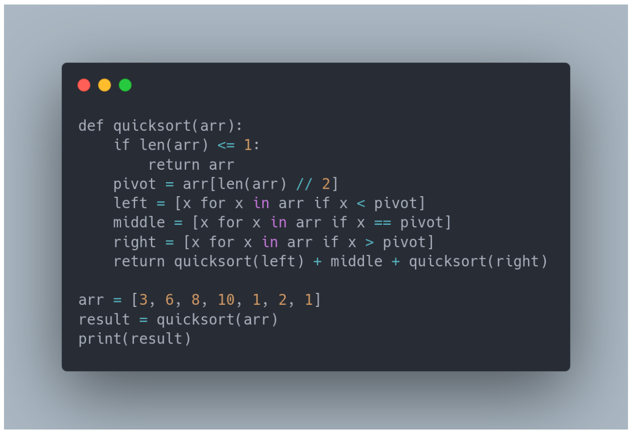
```
[11, 12, 22, 25, 34, 64, 90]
```



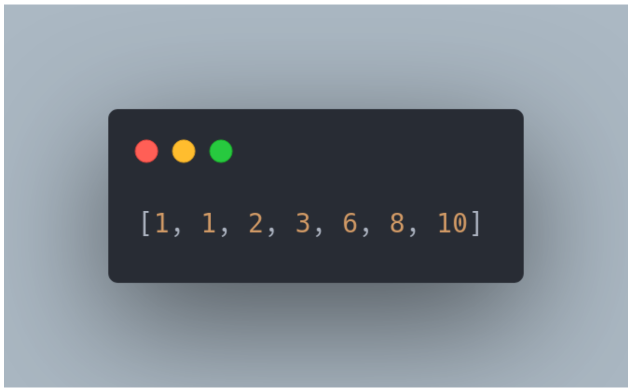
```python
def caesar_cipher(text, shift):
    encrypted = ""
    for char in text:
        if char.isalpha():
            shifted = chr(((ord(char) - ord('a' if char.islower() else 'A') + shift) % 26) + ord('a' if
char.islower() else 'A'))
            encrypted += shifted
        else:
            encrypted += char
    return encrypted

result = caesar_cipher("Hello, World!", 3)
print(result)
```



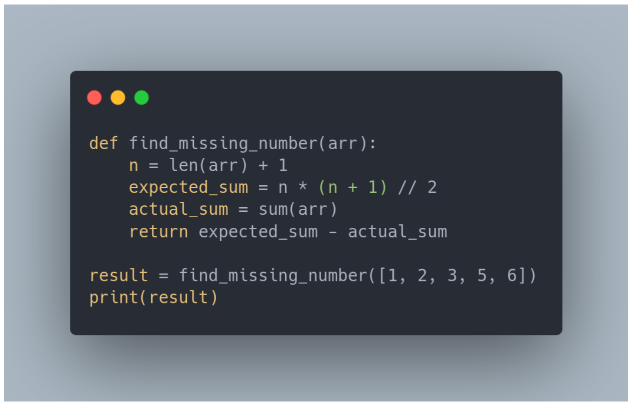
```
Khoor, Zruog!
```



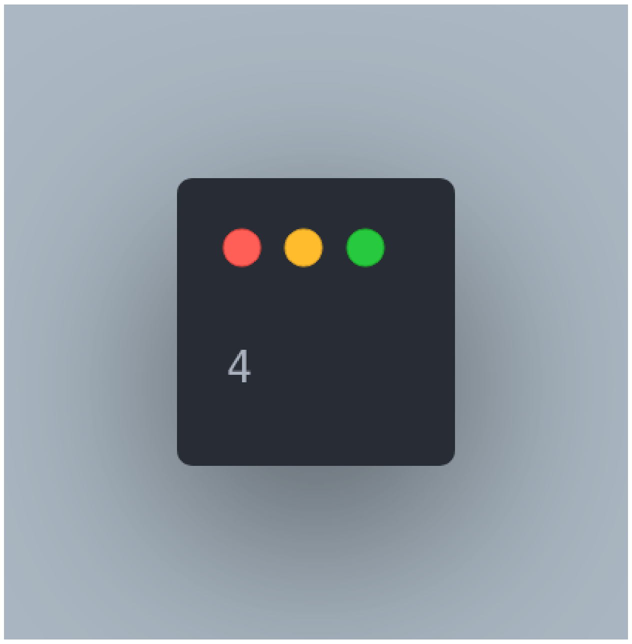
```python
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)

arr = [3, 6, 8, 10, 1, 2, 1]
result = quicksort(arr)
print(result)
```



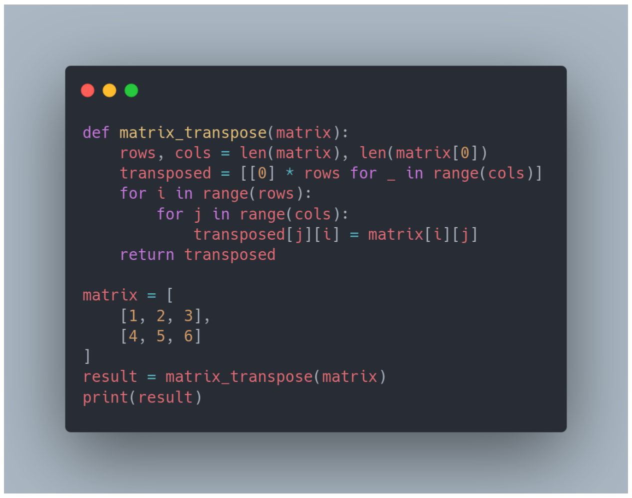
```
[1, 1, 2, 3, 6, 8, 10]
```



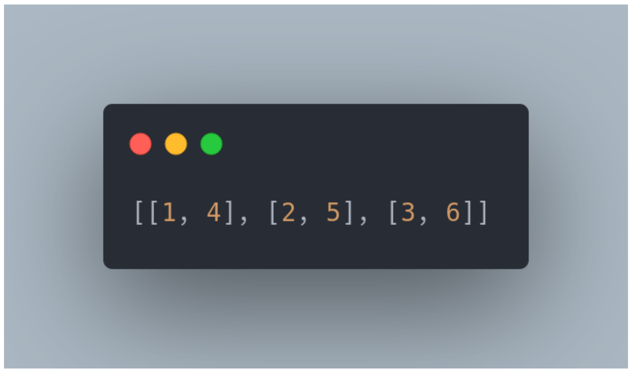
```python
def find_missing_number(arr):
    n = len(arr) + 1
    expected_sum = n * (n + 1) // 2
    actual_sum = sum(arr)
    return expected_sum - actual_sum

result = find_missing_number([1, 2, 3, 5, 6])
print(result)
```



```
4
```



```python
def matrix_transpose(matrix):
    rows, cols = len(matrix), len(matrix[0])
    transposed = [[0] * rows for _ in range(cols)]
    for i in range(rows):
        for j in range(cols):
            transposed[j][i] = matrix[i][j]
    return transposed

matrix = [
    [1, 2, 3],
    [4, 5, 6]
]
result = matrix_transpose(matrix)
print(result)
```



```
[[1, 4], [2, 5], [3, 6]]
```