C. Case Study 3: Amortized Analysis - Dynamic Arrays

6. Write a Python script to simulate the insertion of posts into a dynamic array and calculate the amortized cost

```
class DynamicArray:
    def __init__(self):
        self.array = []
        self.size = 0
        self.capacity = 1
        self.operations = 0

    def insert(self, post):
        if self.size == self.capacity:
            self.resize()

        self.array.append(post)
        self.size += 1
        self.operations += 1

    def resize(self):
        new_capacity = self.capacity * 2
        new_array = [None] * new_capacity
        for i in range(self.size):
            new_array[i] = self.array[i]

        self.array = new_array
        self.capacity = new_capacity
        self.operations += self.size

    def get_amortized_cost(self):
```

```
        return self.operations / self.size if self.size > 0 else 0

dynamic_array = DynamicArray()

posts = ["Post 1", "Post 2", "Post 3", "Post 4", "Post 5", "Post 6"]
for post in posts:
    dynamic_array.insert(post)

print("Array contents:", dynamic_array.array)
print("Total number of operations:", dynamic_array.operations)
print("Amortized cost per insertion:", dynamic_array.get_amortized_cost())
```