Practice Exercise 1: Vectorization

Problem: Conditional Array Modification

```
import numpy as np

# Problem: Given an array, double all elements greater than 5,
# keep others unchanged.

arr = np.array([1, 6, 3, 8, 2])
print("Original array:", arr)
```

Solution 1: Using Loop (Slow)

```
# Loop version (not recommended)
result_loop = []
for i in range(len(arr)):
    if arr[i] > 5:
        result_loop.append(arr[i] * 2)
    else:
        result_loop.append(arr[i])

result_loop = np.array(result_loop)
print("Loop result:", result_loop)
```

Solution 2: Vectorized (Fast)

```
# Vectorized version (recommended)
result = np.where(arr > 5, arr * 2, arr)
print("Vectorized result:", result)

# Alternative vectorized approach
result_alt = arr.copy()
mask = arr > 5
result_alt[mask] = result_alt[mask] * 2
print("Alternative vectorized:", result_alt)
```

Performance Comparison

```
import time

# Test with large array
large_arr = np.random.randint(1, 10, 1000000)

# Loop version
start = time.time()
result_loop = []
for i in range(len(large_arr)):
    if large_arr[i] > 5:
        result_loop.append(large_arr[i] * 2)
    else:
        result_loop.append(large_arr[i])
loop_time = time.time() - start

# Vectorized version
start = time.time()
```

```
result_vec = np.where(large_arr > 5, large_arr * 2, large_arr)
vec_time = time.time() - start

print(f"Loop time: {loop_time:.4f}s")
print(f"Vectorized time: {vec_time:.4f}s")
print(f"Speedup: {loop_time/vec_time:.1f}x")
```

Extended Exercise: Multiple Conditions

```
# More complex: different operations based on value ranges
arr = np.random.randint(0, 100, 20)
print("Original:", arr)

# If < 30: multiply by 2
# If 30-70: keep same
# If > 70: divide by 2

result = np.where(arr < 30, arr * 2,
                  np.where(arr > 70, arr / 2, arr))
print("Result:", result)

# With conditions
def transform(x):
    if x < 30:
        return x * 2
    elif x > 70:
        return x / 2
    else:
        return x

# Vectorize custom function
vectorized_transform = np.vectorize(transform)
result_custom = vectorized_transform(arr)
print("Custom transform:", result_custom)
```