Problem Solution:

```
def fast_copy(source, count):
    # Allocate destination
    dest = [None] * count
    i = 0
    n = count // 8        # Number of full groups
    r = count % 8         # Remainder
    # Handle partial group using explicit control flow (Duff-style fall-through)
    if r == 7:
        dest[i] = source[i]; i += 1
    if r >= 6:
        dest[i] = source[i]; i += 1
    if r >= 5:
        dest[i] = source[i]; i += 1
    if r >= 4:
        dest[i] = source[i]; i += 1
    if r >= 3:
        dest[i] = source[i]; i += 1
    if r >= 2:
        dest[i] = source[i]; i += 1
    if r >= 1:
        dest[i] = source[i]; i += 1
    # Main loop: exactly eight assignments per iteration
    while n > 0:
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
```

```
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        n -= 1
    return dest
```

Explanation:

```
    def fast copy(source, count):
    Copies 'count' elements from source list to a new destination list
    using a fast copy technique (8 elements per loop iteration).
    Rules followed:
    - Each element is copied one by one (no bulk copy)
    - Remaining elements (<8) are copied first without using a loop
    - Main loop copies exactly 8 elements per iteration
    # 1. Create destination list with required size
    dest = [None] * count
    # 2. Calculate number of full groups of 8
    groups = count // 8
    # 3. Calculate remaining elements
    remainder = count % 8
    # Index variable to track current position
    i = 0
    # 4. Copy remaining elements FIRST (NO LOOP allowed)
    if remainder >= 1:
        dest[i] = source[i]
        i += 1
    if remainder >= 2:
        dest[i] = source[i]
        i += 1
    if remainder >= 3:
        dest[i] = source[i]
        i += 1
```

```
    if remainder >= 4:
        dest[i] = source[i]
        i += 1
    if remainder >= 5:
        dest[i] = source[i]
        i += 1
    if remainder >= 6:
        dest[i] = source[i]
        i += 1
    if remainder >= 7:
        dest[i] = source[i]
        i += 1
    # 5. Main loop: copy exactly 8 elements per iteration
    while groups > 0:
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        dest[i] = source[i]; i += 1
        groups -= 1   # Reduce group count
    # 6. Return copied destination list
    return dest
```

Example

```
source = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
count = 10
result = fast copy(source, count)
```

print(result)

Output

[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Key Learning Points

No loop used for leftover elements
Exactly 8 assignments inside the loop
One loop check for every 8 elements
Each element copied individually

M.Anitha

AP/AI&ML