

```

def first_fit(blocks, processes):
    allocation = [-1] * len(processes)

    for i, process in enumerate(processes):
        for j, block in enumerate(blocks):
            if block >= process:
                allocation[i] = j
                blocks[j] -= process
                break
    return allocation

# Example usage
if __name__ == "__main__":
    blocks = [100, 500, 200, 300, 600]
    processes = [212, 417, 112, 426]

    allocation = first_fit(blocks.copy(), processes)

    print("First Fit Allocation:")
    for i, process in enumerate(processes):
        if allocation[i] != -1:
            print(f"Process {i+1} of size {process} allocated to Block {allocation[i]+1}")
        else:
            print(f"Process {i+1} of size {process} cannot be allocated")

```

OUTPUT:

First Fit Allocation:

Process 1 of size 212 allocated to Block 2

Process 2 of size 417 allocated to Block 5

Process 3 of size 112 allocated to Block 2

Process 4 of size 426 allocated to Block 5

```

def best_fit(blocks, processes):
    allocation = [-1] * len(processes)

    for i, process in enumerate(processes):
        best_idx = -1
        min_space = float('inf')
        for j, block in enumerate(blocks):
            if block >= process and (block - process) < min_space:
                best_idx = j
                min_space = block - process
        if best_idx != -1:
            allocation[i] = best_idx
            blocks[best_idx] -= processes[i]

    return allocation

# Example usage
if __name__ == "__main__":
    blocks = [100, 500, 200, 300, 600]
    processes = [212, 417, 112, 426]

    allocation = best_fit(blocks.copy(), processes)

    print("Best Fit Allocation:")
    for i, process in enumerate(processes):
        if allocation[i] != -1:
            print(f"Process {i+1} of size {process} allocated to Block {allocation[i]+1}")
        else:
            print(f"Process {i+1} of size {process} cannot be allocated")

```

OUTPUT:

Best Fit Allocation:

Process 1 of size 212 allocated to Block 2

Process 2 of size 417 allocated to Block 5

Process 3 of size 112 allocated to Block 3

Process 4 of size 426 allocated to Block 5

```
def next_fit(blocks, processes):  
    allocation = [-1] * len(processes)  
    last_alloc = 0  
  
    for i, process in enumerate(processes):  
        j = last_alloc  
        while True:  
            if blocks[j] >= process:  
                allocation[i] = j  
                blocks[j] -= process  
                last_alloc = j  
                break  
            j = (j + 1) % len(blocks)  
        if j == last_alloc:  
            break  
  
    return allocation  
  
# Example usage  
if __name__ == "__main__":  
    blocks = [100, 500, 200, 300, 600]  
    processes = [212, 417, 112, 426]  
  
    allocation = next_fit(blocks.copy(), processes)
```

```

print("Next Fit Allocation:")
for i, process in enumerate(processes):
    if allocation[i] != -1:
        print(f"Process {i+1} of size {process} allocated to Block {allocation[i]+1}")
    else:
        print(f"Process {i+1} of size {process} cannot be allocated")

```

OUTPUT:

Next Fit Allocation:

Process 1 of size 212 allocated to Block 2

Process 2 of size 417 allocated to Block 5

Process 3 of size 112 allocated to Block 2

Process 4 of size 426 allocated to Block 5

```

def worst_fit(blocks, processes):
    allocation = [-1] * len(processes)

    for i, process in enumerate(processes):
        worst_idx = -1
        max_space = -1
        for j, block in enumerate(blocks):
            if block >= process and (block - process) > max_space:
                worst_idx = j
                max_space = block - process
        if worst_idx != -1:
            allocation[i] = worst_idx
            blocks[worst_idx] -= processes[i]

    return allocation

```

```

# Example usage

if __name__ == "__main__":
    blocks = [100, 500, 200, 300, 600]
    processes = [212, 417, 112, 426]

    allocation = worst_fit(blocks.copy(), processes)

    print("Worst Fit Allocation:")
    for i, process in enumerate(processes):
        if allocation[i] != -1:
            print(f"Process {i+1} of size {process} allocated to Block {allocation[i]+1}")
        else:
            print(f"Process {i+1} of size {process} cannot be allocated")

```

OUTPUT:

Worst Fit Allocation:

Process 1 of size 212 allocated to Block 5

Process 2 of size 417 allocated to Block 2

Process 3 of size 112 allocated to Block 5

Process 4 of size 426 allocated to Block 5