

Write a program that implements the LRU page-replacement algorithm.

```
from collections import deque
```

```
def lru_page_replacement(page_references, frame_size):
```

```
    """
```

```
        Implements the LRU page replacement algorithm.
```

```
    Args:
```

```
    page_references: A list of page references.
```

```
    frame_size: The number of frames available.
```

```
    Returns:
```

```
    The number of page faults.
```

```
    """
```

```
    frames = deque(maxlen=frame_size)
```

```
    page_faults = 0
```

```
    page_index = {} # Dictionary to store page indices
```

```
    for page in page_references:
```

```
        if page not in page_index:
```

```
            page_faults += 1
```

```
            if len(frames) == frame_size:
```

```
                lru_page = frames.popleft()
```

```
                del page_index[lru_page]
```

```
            frames.append(page)
```

```
            page_index[page] = len(frames) - 1
```

```
        else:
```

```
            frames.rotate(-page_index[page])
```

```
    return page_faults
```

```
# Example usage
```

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

```
frame_size = 3
```

```
page_faults = lru_page_replacement(page_references, frame_size)
```

```
print("Number of page faults:", page_faults)
```

```
from collections import deque
```

```
def lru_page_replacement(page_references, frame_size):
```

```
    """
```

```
        Implements the LRU page replacement algorithm.
```

```
    Args:
```

```
    page_references: A list of page references.
```

```
    frame_size: The number of frames available.
```

```
    Returns:
```

```
    The number of page faults.
```

```
    """
```

```
frames = deque(maxlen=frame_size)
page_faults = 0
page_index = {} # Dictionary to store page indices
```

```
for page in page_references:
    if page not in page_index:
        page_faults += 1
    if len(frames) == frame_size:
        lru_page = frames.popleft()
        del page_index[lru_page]
    frames.append(page)
    page_index[page] = len(frames) - 1
else:
    frames.rotate(-page_index[page])
```

```
return page_faults
```

Example usage

```
page_references_per_process = {
    1: [1, 2, 3, 1, 4, 5, 1, 2, 3, 4, 5],
    2: [2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2],
    3: [3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3],
    4: [4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4],
    5: [5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5]
}
frame_size = 3
```

```
page_faults_per_process = lru_page_replacement(page_references_per_process, frame_size)
print("Page faults per process:")
for process_id, page_faults in page_faults_per_process.items():
    print(f"Process {process_id}: {page_faults}")
```