Sample input-output results

Cache Size:4

Reference String: a b c a d c a e c

Replacement Policy: LFU

Step	Memory Access	Hit / Miss	Action Taken	Frequency no	Cache Status
1	a	Miss	Cache empty → Insert a	[1,0,0,0]	[a]
2	b	Miss	Insert b	[1,1,0,0]	[a,b]
3	c	Miss	Insert c	[1,1,1,0]	[a,b,c]
4	a	**Hit**	Block already in cache→ frequency increases	[2,1,1,0]	[a,b,c]
5	d	Miss	Insert d	[2,1,1,1]	[a,b,c,d]
6	c	**Hit**	Block already in cache→ frequency increases	[2,1,2,1]	[a,b,c,d]
7	a	**Hit**	Block already in cache→ frequency increases	[3,1,2,1]	[a,b,c,d]
8	e	Miss	Cache full → replace lowest frequency block (b)	[2,1,2,1]	[a,e,c,d]
9	c	Miss	Block already in cache→ frequency increases	[3,1,3,1]	[a,e,c,d]

Output after all the steps (Output of the Program)

```
Enter cache size: 4
Enter access sequence (space separated): a b c a d c a e c

--- LFU Simulation ---

Accessing: a
Status: MISS / FAULT
Cache   : ['a']
Freq    : [1]

Accessing: b
Status: MISS / FAULT
Cache   : ['a', 'b']
Freq    : [1, 1]

Accessing: c
Status: MISS / FAULT
Cache   : ['a', 'b', 'c']
Freq    : [1, 1, 1]

Accessing: a
Status: HIT
Cache   : ['a', 'b', 'c']
Freq    : [2, 1, 1]

Accessing: d
Status: MISS / FAULT
Cache   : ['a', 'b', 'c', 'd']
Freq    : [2, 1, 1, 1]
```

```
Accessing: d
Status: MISS / FAULT
Cache   : ['a', 'b', 'c', 'd']
Freq    : [2, 1, 1, 1]

Accessing: c
Status: HIT
Cache   : ['a', 'b', 'c', 'd']
Freq    : [2, 1, 2, 1]

Accessing: a
Status: HIT
Cache   : ['a', 'b', 'c', 'd']
Freq    : [3, 1, 2, 1]

Accessing: e
Status: MISS / FAULT
Removing (LFU): b
Cache   : ['a', 'e', 'c', 'd']
Freq    : [3, 1, 2, 1]

Accessing: c
Status: HIT
Cache   : ['a', 'e', 'c', 'd']
Freq    : [3, 1, 3, 1]

--- SUMMARY ---
Total Hits  : 4
Total Misses: 5
Hit Ratio   : 0.4444
```