main.c

Run

Output

Clear

```c
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>

// SHA-3 state: 5x5 matrix of 64-bit lanes (25 lanes total)
#define LANES 25
#define RATE_LANES 16
#define CAPACITY_LANES 9

// Function to check if all capacity lanes are nonzero
bool all_capacity_nonzero(uint64_t state[]) {
    for (int i = RATE_LANES; i < LANES; i++) {
        if (state[i] == 0)
            return false;
```

```
All capacity lanes became nonzero after 1 message block (s).


=== Code Execution Successful ===
```

main.c

Run

Output

Clear

```
44
45          // Simulate diffusion to capacity lanes:
                XOR some bits from rate into
                capacity
46          for (int i = 0; i < RATE_LANES; i++) {
47              int target = RATE_LANES + (i %
                    CAPACITY_LANES);
48              state[target] ^= state[i] &
                    (((uint64_t)rand() << 32) | rand
                    ()); // simulate influence
49          }
50      }
51
52      printf("All capacity lanes became nonzero
            after %d message block(s).\n", rounds);
53      return 0;
54  }
55
```

```
All capacity lanes became nonzero after 1 message block
    (s).


=== Code Execution Successful ===
```