Program29 c program in SHA-12

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <stdint.h>

#define STATE\_SIZE 25

#define CAPACITY\_LANES 16

#define LANE\_SIZE 64

typedef struct {

uint64\_t state[STATE\_SIZE];

} InternalState;

void initializeState(InternalState \*state)

{

for (int i = 0; i < STATE\_SIZE; i++)

{

state->state[i] = 0;

}

}

int allCapacityLanesNonzero(InternalState \*state)

{

for (int i = 0; i < CAPACITY\_LANES; i++)

{

if (state->state[i] == 0)

{

return 0;

}

}

return 1;

}

int main()

{

InternalState state;

initializeState(&state);

srand(time(NULL));

int steps = 0;

while (!allCapacityLanesNonzero(&state))

{

int laneToUpdate = rand() % CAPACITY\_LANES;

int bitPosition = rand() % LANE\_SIZE;

state.state[laneToUpdate] |= (1ULL << bitPosition);

steps++;

}

printf("All capacity lanes have at least one nonzero bit after %d steps.\n", steps);

return 0;

}

OUTPUT:

All capacity lanes have at least one nonzero bit after 60 steps.