20) Write a python program for SHA-3 option with a block size of 1024 bits and assume that each of the lanesin the first message block (P0) has at least one nonzero bit. To start, all of the lanes in the internal statematrix that correspond to the capacity portion of the initial state are all zeros. Show how long it will takebefore all of these lanes have at least one nonzero bit. Note: Ignore the permutation. That is, keep trackof the original zero lanes even after they have changed position in the matrix.

PROGRAM:-

```
import random


def simulate_sha3_spread():
    total_lanes = 25
    lane_size = 64  # bits
    capacity_lanes = 9  # last 9 lanes
    state = [1] * (total_lanes - capacity_lanes) + [0] * capacity_lanes

    steps = 0
    while 0 in state[-capacity_lanes:]:
        new_state = state.copy()

        # Simulate mixing: each lane is XORed with two random other lanes
        for i in range(total_lanes):
            a, b = random.sample(range(total_lanes), 2)
            new_state[i] ^= state[a] | state[b]

        state = new_state
        steps += 1

    return steps

# Run simulation multiple times to get average
runs = 20
results = [simulate_sha3_spread() for _ in range(runs)]
average_steps = sum(results) / runs
```

```
print(f"Average steps until all capacity lanes are non-zero (over {runs} runs): {average_steps:.2f}")

print("Individual runs:", results)
```

OUTPUT:-

```
Average steps until all capacity lanes are non-zero (over 20 runs): 543.80

Individual runs: [1669, 1, 1294, 575, 778, 1231, 457, 984, 963, 545, 207, 588, 1, 59, 1, 128, 1, 1, 221, 1172]
```