**ASSIGNMENT 3:**

Question 5: Technical Debt Reduction Write a C program to simulate a technical debt reduction program in Amazon's platform. Assume that the platform has 1 million lines of code and each line of code has a technical debt of 0.1 (10%). Write a program to simulate the reduction of technical debt over time and display the result.

#include <stdio.h>

#define TOTAL\_LINES 1000000

#define INITIAL\_DEBT\_PERCENT 0.1

#define DEBT\_REDUCTION\_RATE 0.05

#define DEBT\_THRESHOLD 1000

int main()

{

float technical\_debt = TOTAL\_LINES \* INITIAL\_DEBT\_PERCENT;

int iterations = 0;

printf("Starting Technical Debt: %.f lines\n", technical\_debt);

while (technical\_debt > DEBT\_THRESHOLD)

{

technical\_debt -= technical\_debt \* DEBT\_REDUCTION\_RATE;

iterations++;

printf("Iteration %d: Remaining Debt = %.2f lines\n", iterations, technical\_debt);

}

printf("Technical debt reduced to acceptable levels in %d iterations.\n",iterations);

return 0;

}

Write a C program to optimize the logistics and supply chain of Flipkart's platform. Assume that the platform has 5 warehouses and 10 delivery locations. Write a program to find the shortest path between each warehouse and delivery location using Dijkstra's algorithm.

#include <stdio.h>

#include <limits.h>

#define WAREHOUSES 5

#define LOCATIONS 10

#define N (WAREHOUSES + LOCATIONS)

int minDistance(int dist[], int visited[])

{

int min = INF, min\_index = -1;

for (int v = 0; v < N; v++) {

if (!visited[v] && dist[v] <= min)

{

min = dist[v];

min\_index = v;

}

}

return min\_index;

}

void dijkstra(int graph[N][Nnt dist[N];

int visited[N] = {0};

for (int i = 0; i < N; i++)

{

dist[i] = INF;

}

dist[src] = 0;

for (int count = 0; count < N - 1; count++)

{

int u = minDistance(dist, visited);

visited[u] = 1;

for (int v = 0; v < N; v++)

{

if (!visited[v] && graph[u][v] && dist[u] != INF && dist[u] + graph[u][v] < dist[v])

{

dist[v] = dist[u] + graph[u][v];

}

}

}

printf("\nShortest distances from Warehouse %d:\n", src + 1);

for (int i = WAREHOUSES; i < N; i++) {

printf("To Location %d: %d\n", i - WAREHOUSES + 1, dist[i]);

}

}

int main()

{

for (int i = 0; i < WAREHOUSES; i++) { dijkstra(graph, i);

return 0;

}

Write a C program to simulate a scalability issue in Amazon's platform. Assume that the platform can handle a maximum of 500,000 concurrent users. If the number of users exceeds 500,000, the platform crashes. Write a program to simulate this scenario and display the result.

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#define MAX\_USERS 500000

int main()

{

int current\_users = 0;

printf("Simulating Amazon's platform scalability issue...\n");

while (1)

{

current\_users += rand() % 50000;

printf("Current Users: %d\n", current\_users);

if (current\_users > MAX\_USERS)

{

printf("ERROR: Platform crashed! Too many users connected.\n");

exit(EXIT\_FAILURE);

}

sleep(1);

}

return 0;

}

**GITHUB LINK:**