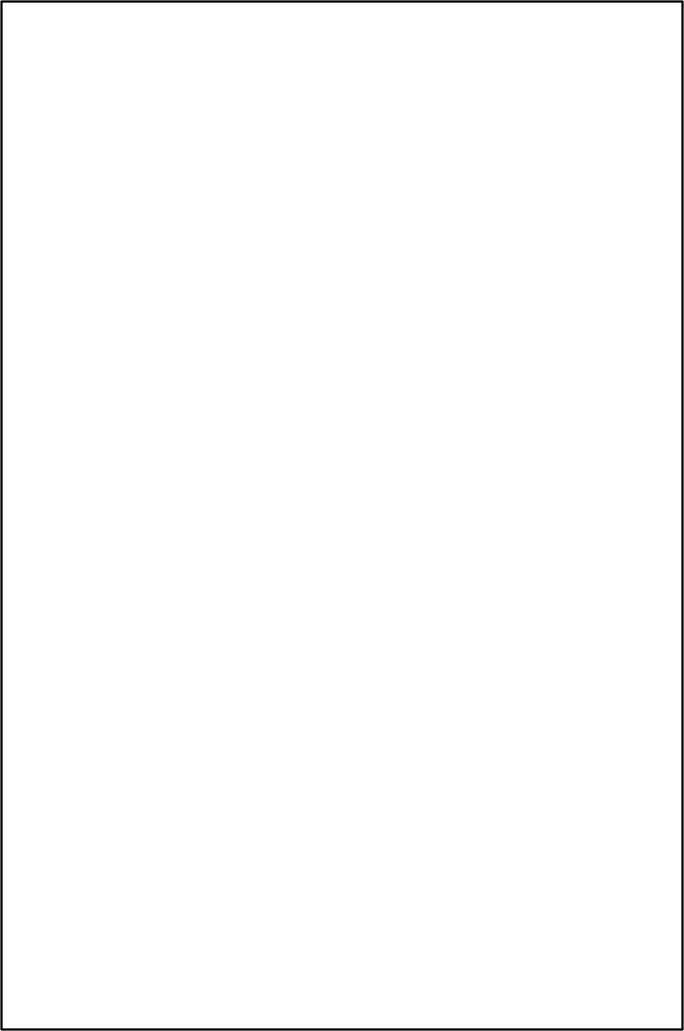
Data Structures Assignment 3



Name :

**PRN** : B24CE1047

**Branch and Batch** : Computer Engineering SY1 Batch (B)



**Title** : load balancing

**Problem statement** :

Imagine you have a set of servers that handle requests for a web application. The key to load balancing is using the hash value of a client ip address or a request ID to determine which server should handle the request . The hash function is typically designed by which server should handle the request . The hash function is typically designed so that the data is evenly distributed across the servers , ensuring that no single server is overloaded . write a program of a load balancing system



**CODE :**

#include <iostream>

using namespace std;

const int Servers = 10;

int hashvalue(int requestID) {

return requestID % Servers;

}

int main() {

cout<<"Servers = 10 "<<endl;

int n;

cout << "Enter the number of requests: ";

cin >> n;

int request[100];

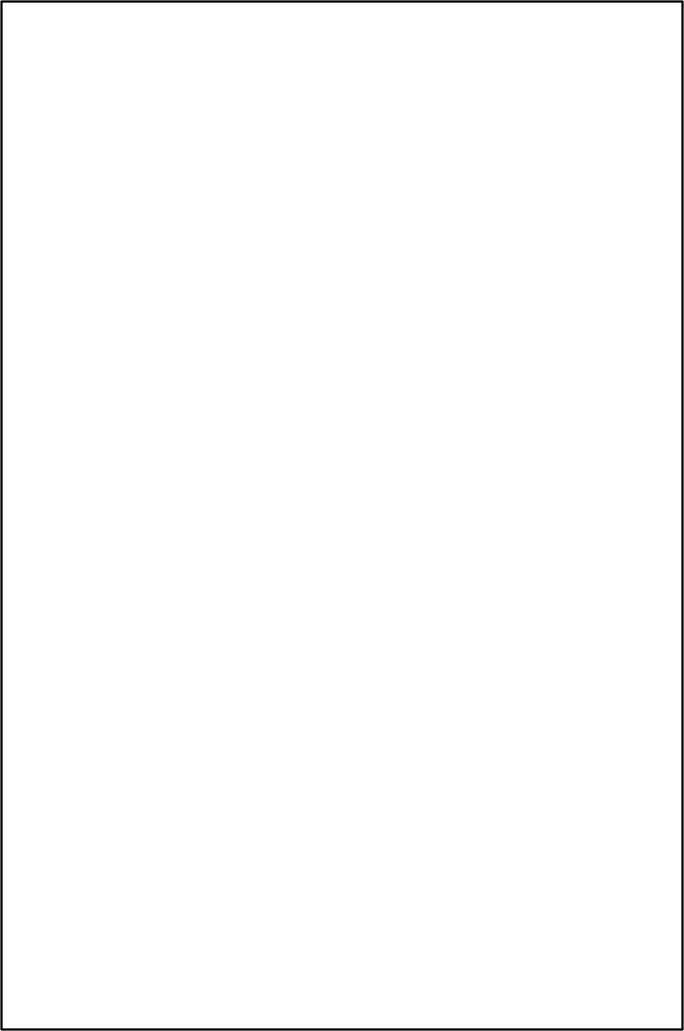
int hash\_server[Servers];

for (int i = 0; i < Servers; i++) {

hash\_server[i] = -1;

}

for (int i = 0; i < n; i++) {

 cout << "Enter request ID " << i + 1 << ": ";

cin >> request[i];

int hash\_val = hashvalue(request[i]);

if (hash\_server[hash\_val] == -1) {

hash\_server[hash\_val] = request[i];

cout << "Request " << request[i] << " is assigned to Server " << hash\_val << endl;

} else {

cout << "Collision " << request[i] << " at Server " << hash\_val;

int original = hash\_val;

int i = 0;

bool assigned = false;

while (i < Servers) {

hash\_val = (original + i) % Servers;

if (hash\_server[hash\_val] == -1) {

hash\_server[hash\_val] = request[i];

cout << "Request " << request[i] << " is assigned to Server " << hash\_val << " \n";

assigned = true;

break;

}

i++;

}

if (!assigned) {

cout << "All servers are full! Request " << request[i] << " cannot be assigned.\n";

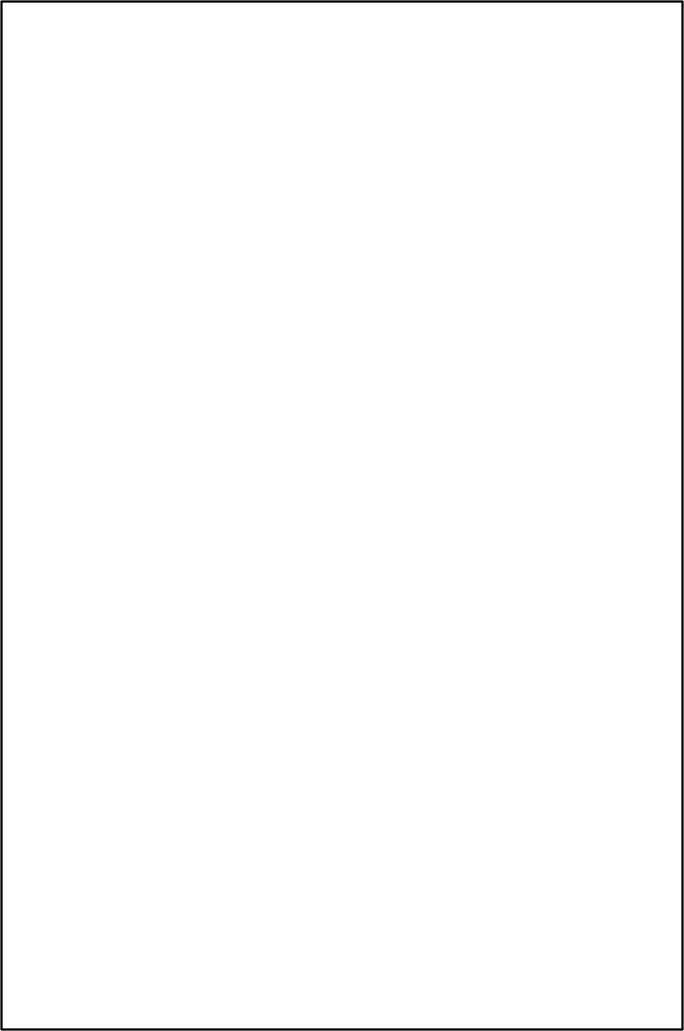
}

}

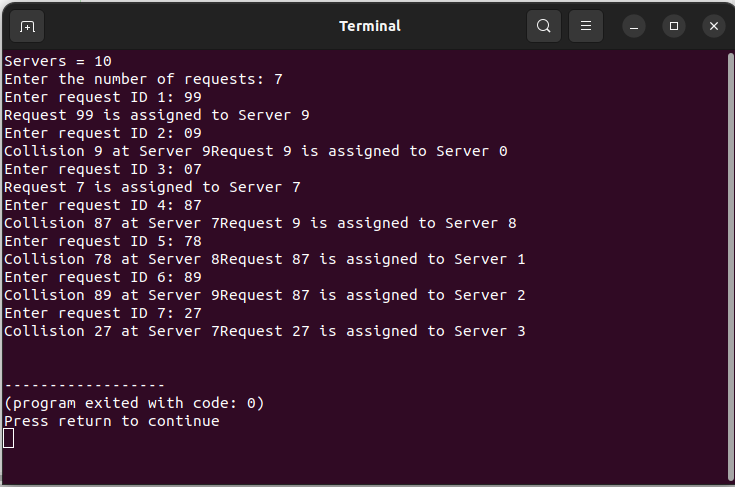
}

return 0;

}

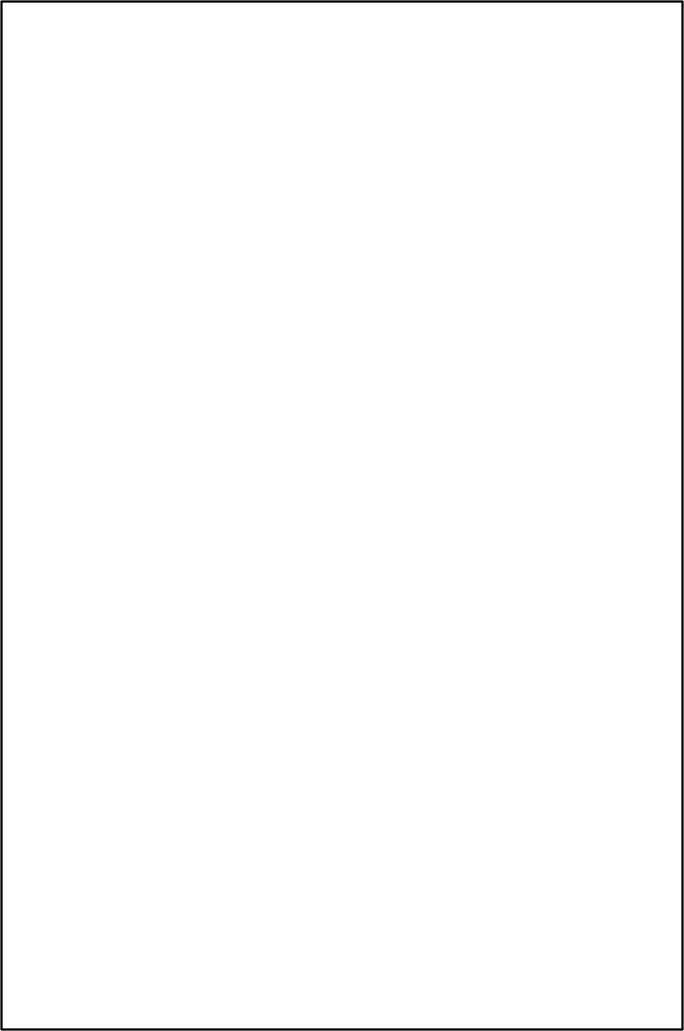


**OUTPUT :**





**THANK YOU !!!**

****