Name: Ghodake Omkar Atul

Roll No: 30

PRN No: 72036196G

BE IT

**Assignment 6**

**Bully and Ring Algorithm for Leader Election**

**Code:**

#include <iostream>

#include <vector>

#include <algorithm>

struct Pro {

int id;

bool act;

Pro(int id) {

this->id = id;

act = true;

}

};

class myc {

public:

int TotalProcess;

std::vector<Pro> process;

GFG() {}

void initialisemyc() {

std::cout << "No of processes 5" << std::endl;

TotalProcess = 5;

process.reserve(TotalProcess);

for (int i = 0; i < process.capacity(); i++) {

process.emplace\_back(i);

}

}

void Election() {

std::cout << "Process no " << process[FetchMaximum()].id << " fails" << std::endl;

process[FetchMaximum()].act = false;

std::cout << "Election Initiated by 2" << std::endl;

int initializedProcess = 2;

int old = initializedProcess;

int newer = old + 1;

while (true) {

if (process[newer].act) {

std::cout << "Process " << process[old].id << " pass Election(" << process[old].id << ") to" << process[newer].id << std::endl;

old = newer;

}

newer = (newer + 1) % TotalProcess;

if (newer == initializedProcess) {

break;

}

}

std::cout << "Process " << process[FetchMaximum()].id << " becomes coordinator" << std::endl;

int coord = process[FetchMaximum()].id;

old = coord;

newer = (old + 1) % TotalProcess;

while (true) {

if (process[newer].act) {

std::cout << "Process " << process[old].id << " pass Coordinator(" << coord << ") message to process " << process[newer].id << std::endl;

old = newer;

}

newer = (newer + 1) % TotalProcess;

if (newer == coord) {

std::cout << "End Of Election " << std::endl;

break;

}

}

}

int FetchMaximum() {

int Ind = 0;

int maxId = -9999;

for (int i = 0; i < process.size(); i++) {

if (process[i].act && process[i].id > maxId) {

maxId = process[i].id;

Ind = i;

}

}

return Ind;

}

};

int main() {

myc object;

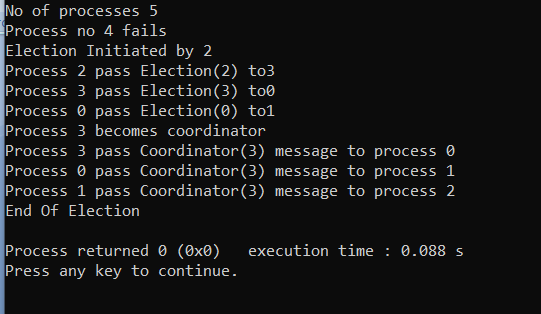
object.initialisemyc();

object.Election();

return 0;

}

**Output:**

****