**NAME: MUZZAMMIL ALI KHAN**

**CLASS : BS(CS) 3-A**

**ROLL NO: 02-134231-025**

1. Design two different algorithms with different approaches to solve the above problem.

**ANSWER:**

**Algorithm 1:**

* Create a circular linked list representing the circle of people.
* Start with person 1 holding the hot potato.
* Iterate through the linked list, passing the hot potato from one person to the next.
* After M passes, remove the person holding the hot potato from the list.
* Continue this process until only one person remains.

**Algorithm 2:**

* Create an array representing the circle of people.
* Start with person 1 holding the hot potato.
* Simulate the passing of the hot potato by incrementing the index of the person in the array.
* After M passes, remove the person holding the hot potato from the array.
* Continue this process until only one person remains.

b)

**CODE:**

#include <iostream>

#include <list>

#include <queue>

using namespace std;

int lastPersonStanding1(int N, int M) {

list<int> circle;

for (int i = 1; i <= N; ++i) {

circle.push\_back(i);

}

auto it = circle.begin();

while (circle.size() > 1) {

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

if (it == circle.end()) {

it = circle.begin();

}

++it;

if (it == circle.end()) {

it = circle.begin();

}

}

it = circle.erase(it);

}

return circle.front();

}

int lastPersonStanding2(int N, int M) {

queue<int> circle;

for (int i = 1; i <= N; ++i) {

circle.push(i);

}

while (circle.size() > 1) {

for (int i = 0; i < M - 1; ++i) {

int front = circle.front();

circle.pop();

circle.push(front);

}

circle.pop();

}

return circle.front();

}

int main() {

int M, N;

N = 5;

M = 1;

cout << "Winner for N=" << N << ", M=" << M << " using Algorithm 1: " << lastPersonStanding1(N, M) << endl;

cout << "Winner for N=" << N << ", M=" << M << " using Algorithm 2: " << lastPersonStanding2(N, M) << endl;

N = 5;

M = 2;

cout << "Winner for N=" << N << ", M=" << M << " using Algorithm 1: " << lastPersonStanding1(N, M) << endl;

cout << "Winner for N=" << N << ", M=" << M << " using Algorithm 2: " << lastPersonStanding2(N, M) << endl;

N = 10;

M = 3;

cout << "Winner for N=" << N << ", M=" << M << " using Algorithm 1: " << lastPersonStanding1(N, M) << endl;

cout << "Winner for N=" << N << ", M=" << M << " using Algorithm 2: " << lastPersonStanding2(N, M) << endl;

return 0;

}

**OUTPUT:**

