ASSIGNMENT 02

Marks: 05

Submission Date: 26-April-2024

NAME:	Abdullah
CLASS:	BS (CS) 3A
REG #:	02-131222-099

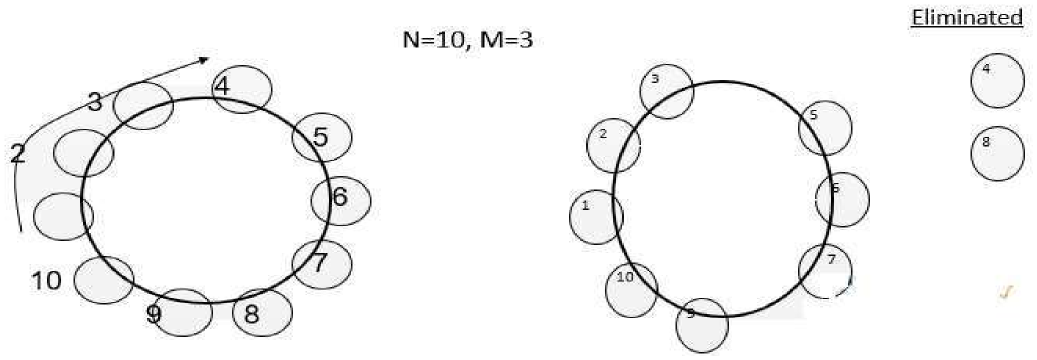
Marks Obtained:____________________

Instructions.

1. Follow same format for assignment submission.
2. Copied/Plagiarized answers will be marked zero.
3. Output must be attached with the code.

Scenario **CLO4, PLO4, C6**

N people, numbered 1 to N are sitting in a circle. Starting at person 1, a hot potato is passed. After M passes the person holding the hot potato is eliminated, the circle closes ranks, and the game continues with the person who was sitting after the eliminated person picking up the hot potato. The last remaining person wins. Thus, if M=0 and N=5, players are eliminated in order and player 5 wins. If M=1 and N=5 the order of elimination is 2,4,1,5



Question

1. **Design** two different algorithms with different approaches to solve the above problem.
2. **Create** a code in C++ to implement both algorithms by applying appropriate data structure for general values of M and N.

   Test your program with three different values of M and N.

Solution:

Algorithm:

```
1. Function hotPotato(N, M)
    Create an empty circular array circle of size N
    Initialize a variable currentPosition to 0
    For each person i from 1 to N:
        Set circle[i-1] = i
    While circle is not empty:
        currentPosition = (currentPosition + M - 1) % circle.size()
        Remove the person at position currentPosition from circle
  Return circle.front() (the winner)
```

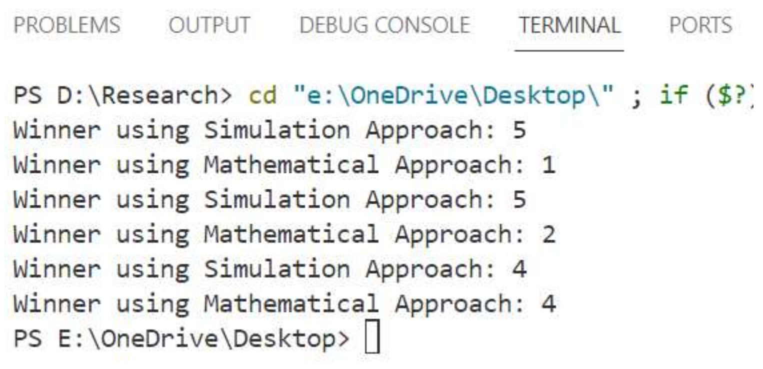
```
2. Function hotPotatoLinkedList(N, M)
    Create an empty circular linked list circle
    Initialize a variable currentPosition to 1

    For each person i from 1 to N:
        Add person i to the end of the linked list circle

    Set the iterator it to the beginning of circle

    While circle is not empty:
        Move the iterator it forward by M - 1 positions
        Remove the person at the position indicated by it
        If the iterator reaches the end of the list, wrap around to the beginning
        Set currentPosition to the person after the removed person
    Return currentPosition (the winner)
```

Code:

```
#include <iostream>
#include <list>
using namespace std;

// Algorithm 1: Simulation Approach
int simulateHotPotato(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 count = 1; count < M; count++) {
            it++;
            if (it == circle.end()) {
                it = circle.begin();
            }
        }
```

```
            it = circle.erase(it);
            if (it == circle.end()) {
                it = circle.begin();
            }
        }

        return circle.front();
}

// Algorithm 2: Mathematical Approach
int calculateWinnerPosition(int N, int M) {
    return (M % N) + 1;
}

int main() {
    int M, N;

    // Test case 1
    M = 0;
    N = 5;
    cout << "Winner using Simulation Approach: " << simulateHotPotato(N, M) <<
endl;
    cout << "Winner using Mathematical Approach: " <<
calculateWinnerPosition(N, M) << endl;

    // Test case 2
    M = 1;
    N = 5;
    cout << "Winner using Simulation Approach: " << simulateHotPotato(N, M) <<
endl;
    cout << "Winner using Mathematical Approach: " <<
calculateWinnerPosition(N, M) << endl;

    // Test case 3
    M = 3;
    N = 7;
    cout << "Winner using Simulation Approach: " << simulateHotPotato(N, M) <<
endl;
    cout << "Winner using Mathematical Approach: " <<
calculateWinnerPosition(N, M) << endl;

    return 0;
}
```

Output:



```
PS D:\Research> cd "e:\OneDrive\Desktop\" ; if ($?
Winner using Simulation Approach: 5
Winner using Mathematical Approach: 1
Winner using Simulation Approach: 5
Winner using Mathematical Approach: 2
Winner using Simulation Approach: 4
Winner using Mathematical Approach: 4
PS E:\OneDrive\Desktop>
```