

Design and Analysis of Algorithms Assignment - 4

Name: Dhanraj Kore

Div: TY B

Roll No: 60

Batch : B-3

Josephus Problem

Approach 1: Using Circular Linked List

CODE :

```
#include <bits/stdc++.h>

using namespace std;

struct Node
{
    int data;

    int b;

    Node *next;
};

Node *insert(Node *node, int x)
{
    Node *temp = new Node;

    temp->data = x;

    temp->b = 0;

    temp->next = NULL;

    if (!node)
    {
        node = temp;

        temp->next = node;

        return node;
    }
}
```

```

Node *p = node;

while (p->next != node)
{
    p = p->next;
}

p->next = temp;
temp->next = node;

return node;
}

Node *create(int n, Node *node)
{
    for (int i = 1; i <= n; i++)
        node = insert(node, i);

    return node;
}

int count_alive(Node *node)
{
    Node *p = node;

    int cnt = 0;

    while (p->next != node)
    {
        if (p->b == 0)
            cnt++;

        p = p->next;
    }

    if (p->b == 0)
        cnt++;

    return cnt;
}

int josephous(Node *node)
{
    Node *temp = node;

    while (count_alive(node) != 1)
    {
        int cnt = 0;

        while (cnt != 1)

```

```

    {
        if (temp->b == 0)
        {
            temp = temp->next;

            cnt++;
        }
        else
            temp = temp->next;
    }

    while (temp->b == 1)
        temp = temp->next;

    temp->b = 1;
}

Node *p = node;
while (p->next != node)
{
    if (p->b == 0)
        return p->data;

    p = p->next;
}

if (p->b == 0)
    return p->data;
}

int main()
{
    int n;

    Node *node = new Node;

    node = NULL;

    cout << "Enter the number: ";

    cin >> n;

    node = create(n, node);

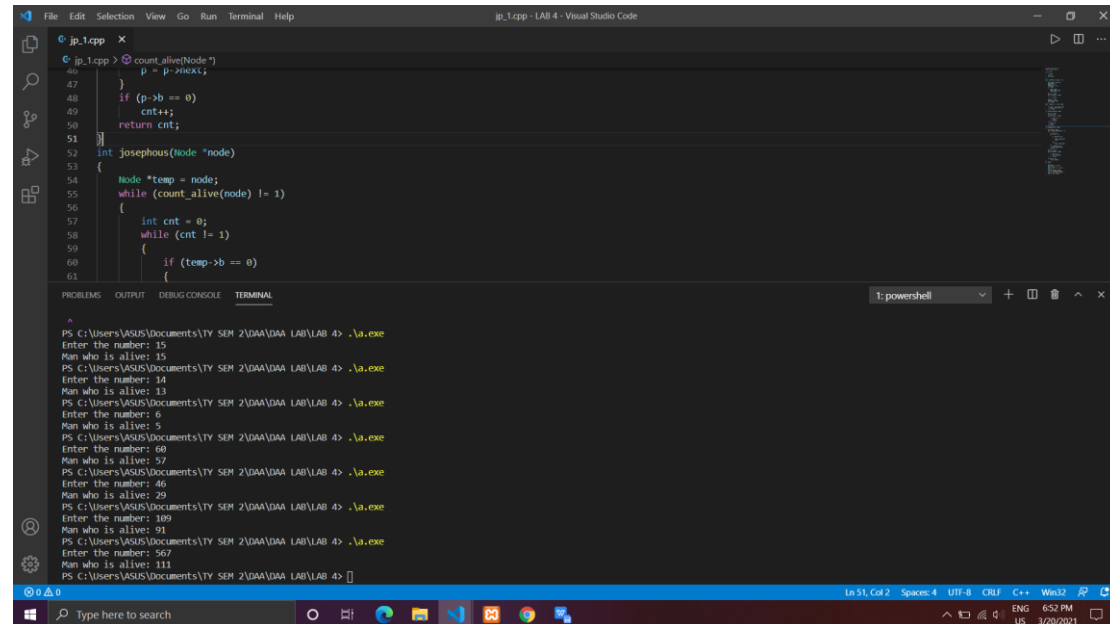
    int ans = josephous(node);

    cout << "Man who is alive: ";

    cout << ans << endl;
}

```

O/P:



The screenshot shows a Visual Studio Code editor with a C++ file named `jp_1.cpp`. The code implements a linked list and a function `count_alive(Node*)` to solve the Josephus problem. The terminal output shows the execution of the program, where the user enters a number of people (15) and a step (5). The program outputs the sequence of people who are alive and the final survivor (11).

```
1  #include <iostream>
2  using namespace std;
3
4  struct Node
5  {
6      int data;
7      Node* next;
8  };
9
10 Node* createNode(int data)
11 {
12     Node* newNode = new Node;
13     newNode->data = data;
14     newNode->next = NULL;
15     return newNode;
16 }
17
18 Node* insertAtEnd(Node* head, int data)
19 {
20     if (head == NULL)
21         return createNode(data);
22     Node* temp = head;
23     while (temp->next != NULL)
24         temp = temp->next;
25     temp->next = createNode(data);
26     return head;
27 }
28
29 Node* count_alive(Node* head)
30 {
31     if (head == NULL)
32         return 0;
33     if (head->next == head)
34         return 1;
35     int cnt = 0;
36     Node* temp = head;
37     while (temp->next != head)
38     {
39         cnt++;
40         temp = temp->next;
41     }
42     return cnt + 1;
43 }
44
45 int josephus(Node* head, int k)
46 {
47     if (head == NULL)
48         return 0;
49     if (head->next == head)
50         return head->data;
51     int cnt = 0;
52     Node* temp = head;
53     while (temp->next != head)
54     {
55         cnt++;
56         temp = temp->next;
57     }
58     if (cnt % k == 0)
59         temp = temp->next;
60     else
61         temp = temp->next->next;
62     return josephus(temp, k);
63 }
64
65 int main()
66 {
67     Node* head = NULL;
68     int n, k;
69     cout << "Enter the number: ";
70     cin >> n;
71     cout << "Man who is alive: ";
72     while (n > 0)
73     {
74         int data;
75         cout << "Enter the number: ";
76         cin >> data;
77         head = insertAtEnd(head, data);
78         n--;
79     }
80     cout << "Man who is alive: ";
81     int k;
82     cout << "Enter the number: ";
83     cin >> k;
84     int ans = josephus(head, k);
85     cout << "Man who is alive: " << ans << endl;
86     return 0;
87 }
```

Terminal Output:

```
PS C:\Users\VASUS\Documents\TY SEM 2\DAI\DAI LAB\LAB 4> .\a.exe
Enter the number: 15
Man who is alive: 15
PS C:\Users\VASUS\Documents\TY SEM 2\DAI\DAI LAB\LAB 4> .\a.exe
Enter the number: 14
Man who is alive: 13
PS C:\Users\VASUS\Documents\TY SEM 2\DAI\DAI LAB\LAB 4> .\a.exe
Enter the number: 6
Man who is alive: 5
PS C:\Users\VASUS\Documents\TY SEM 2\DAI\DAI LAB\LAB 4> .\a.exe
Enter the number: 60
Man who is alive: 57
PS C:\Users\VASUS\Documents\TY SEM 2\DAI\DAI LAB\LAB 4> .\a.exe
Enter the number: 46
Man who is alive: 29
PS C:\Users\VASUS\Documents\TY SEM 2\DAI\DAI LAB\LAB 4> .\a.exe
Enter the number: 109
Man who is alive: 91
PS C:\Users\VASUS\Documents\TY SEM 2\DAI\DAI LAB\LAB 4> .\a.exe
Enter the number: 567
Man who is alive: 111
PS C:\Users\VASUS\Documents\TY SEM 2\DAI\DAI LAB\LAB 4>
```

Approach 2 : Using Recursion

CODE :

```
#include<bits/stdc++.h>

using namespace std;

int josephus(int n)
{
    if(n==0 || n==1)
        return n;

    int ans;

    if(n%2==0)
        ans = 2*josephus(n/2)-1;
    else
        ans = 2*josephus(n/2)+1;

    return ans;
}
```

```

int main()
{
    int n;

    cout<<"Enter the number : ";

    cin>>n;

    int ans = josephus(n);

    cout<<"Man who is alive: "<<ans<<endl;
}

```

O/P:

```

// jp.2.cpp
9      ans = 2*josephus(n/2)-1;
10     else
11         ans = 2*josephus(n/2)+1;
12     return ans;
13 }
14 int main()
15 {
16     int n;
17     cout<<"Enter the number : ";
18     cin>>n;
19     int ans = josephus(n);
20     cout<<"Man who is alive: "<<ans<<endl;
21 }
22

```

```

Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\ASUS\Documents\TY SEM 2\DA\DA LAB\LAB 4> g++ jp.2.cpp
PS C:\Users\ASUS\Documents\TY SEM 2\DA\DA LAB\LAB 4> .\a.exe
Enter the number : 14
Man who is alive: 13
PS C:\Users\ASUS\Documents\TY SEM 2\DA\DA LAB\LAB 4>

```

Approach 2 : Using Bit Manipulation

CODE:

```

#include <bits/stdc++.h>

using namespace std;

int main()

```

```

{

    int n;

    cout << "Enter the number : ";

    cin >> n;

    int m = 1, cnt = 1;

    ;

    while (m < n)

        m = pow(2, cnt++);

    int ans;

    if (n == pow(2, cnt - 1))

        ans = 2 * (n - m) + 1;

    else

    {

        m /= 2;

        ans = 2 * (n - m) + 1;

    }

    cout << "Man who is alive: " << ans << endl;

}

```

O/P:

The screenshot shows the Visual Studio Code interface with the file `jp_3.cpp` open. The code in the editor is as follows:

```

11 while (m < n)
12     m = pow(2, cnt++);
13 int ans;
14 if (n == pow(2, cnt - 1))
15     ans = 2 * (n - m) + 1;
16 else
17 {
18     m /= 2;
19     ans = 2 * (n - m) + 1;
20 }
21 cout << "Man who is alive: " << ans << endl;
22 }
23

```

The terminal window at the bottom shows the execution of the program using `g++ jp_3.cpp` and `./a.exe`. The output is:

```

PS C:\Users\VASUS\Documents\TY_SEH_2\DA\LAB LAB\LAB 4> g++ jp_3.cpp
PS C:\Users\VASUS\Documents\TY_SEH_2\DA\LAB LAB\LAB 4> ./a.exe
Enter the number : 14
Man who is alive: 13
PS C:\Users\VASUS\Documents\TY_SEH_2\DA\LAB LAB\LAB 4>

```