National university of Sciences and Technology

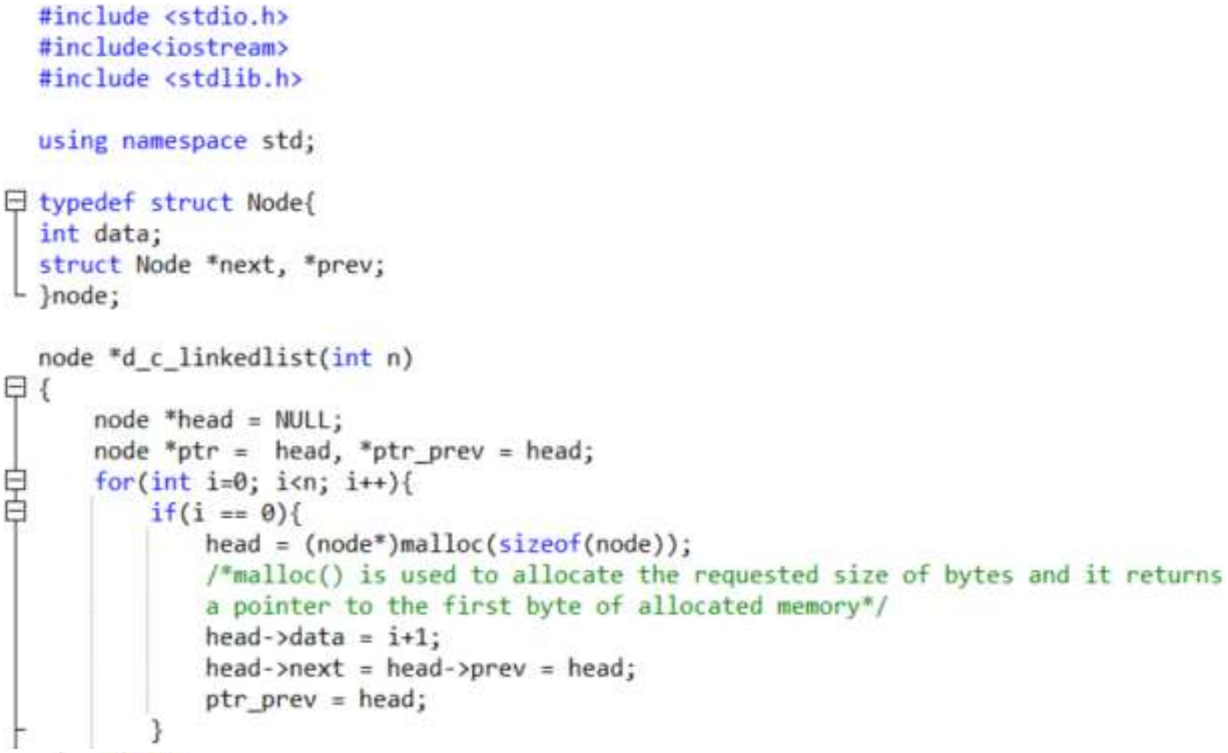
NUST Balochistan Campus (NBC)

Data Structures
Assignment No 2

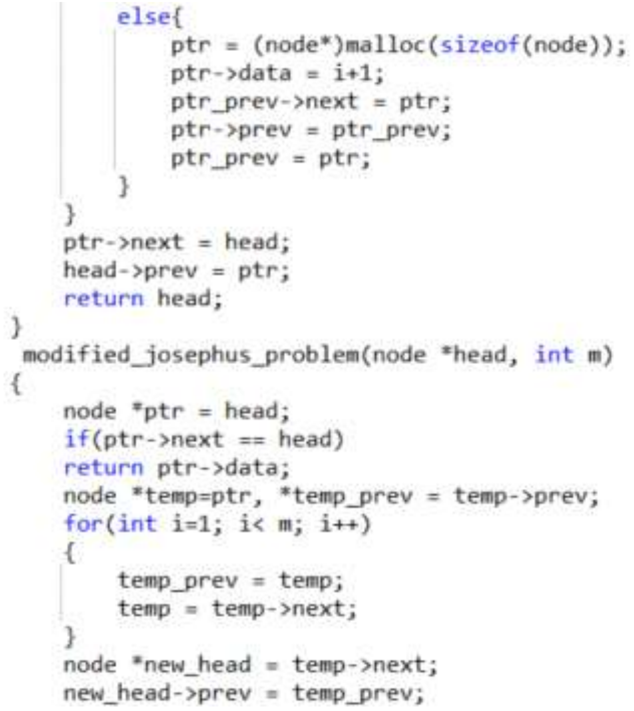
Submitted to: Mr.Najeeb Ullah
Submitted by: Zainab Zahid
Submission Date:24-10-2021
BS(CS) 3rd Semester

Modified Josephus Problem

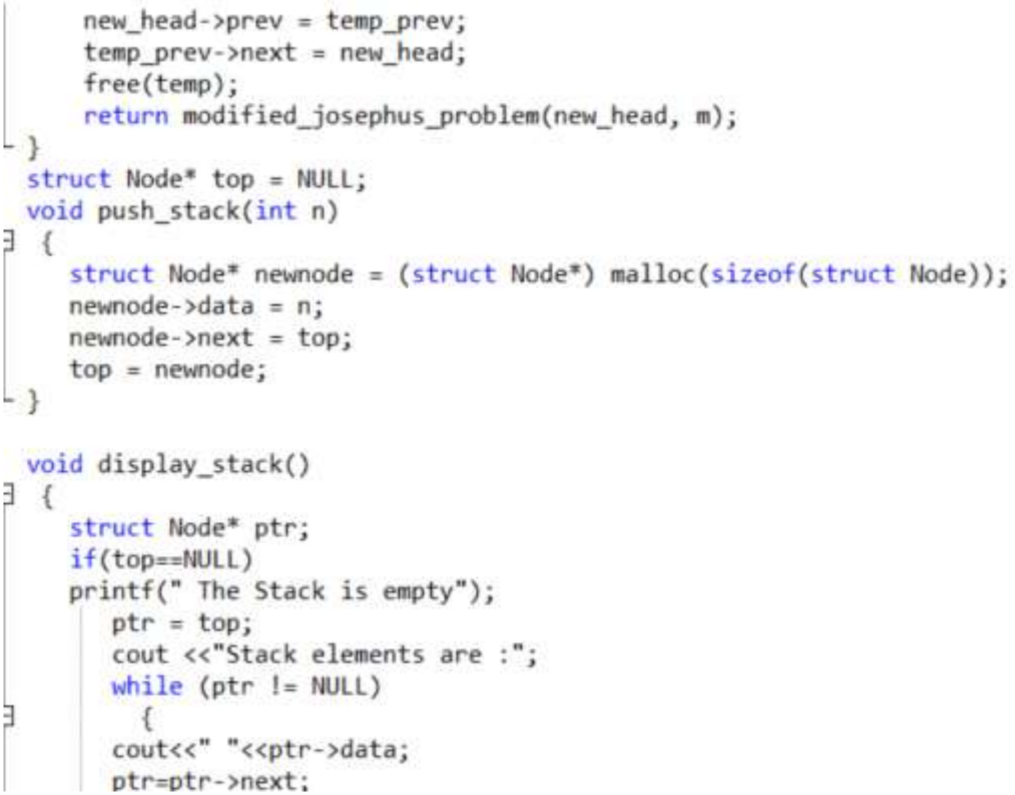
Code:



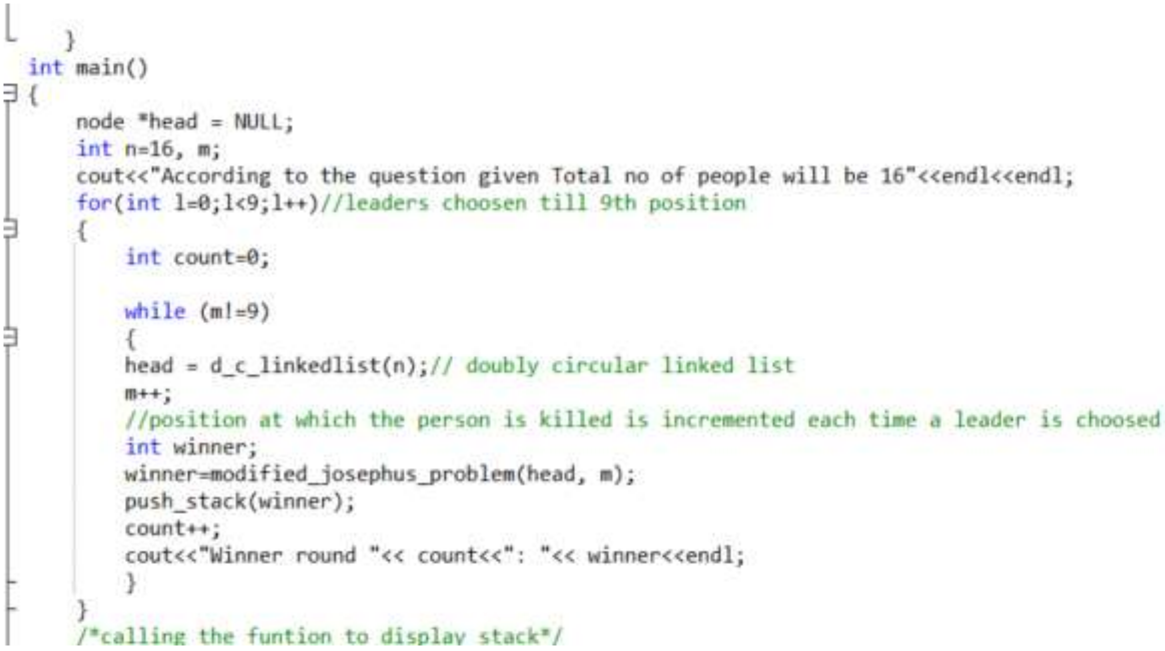
```
#include <stdio.h>
#include<iostream>
#include <stdlib.h>

using namespace std;

typedef struct Node{
int data;
struct Node *next, *prev;
}node;

node *d_c_linkedlist(int n)
{
    node *head = NULL;
    node *ptr =  head, *ptr_prev = head;
    for(int i=0; i<n; i++){
        if(i == 0){
            head = (node*)malloc(sizeof(node));
            /*malloc() is used to allocate the requested size of bytes and it returns
            a pointer to the first byte of allocated memory*/
            head->data = i+1;
            head->next = head->prev = head;
            ptr_prev = head;
        }
```



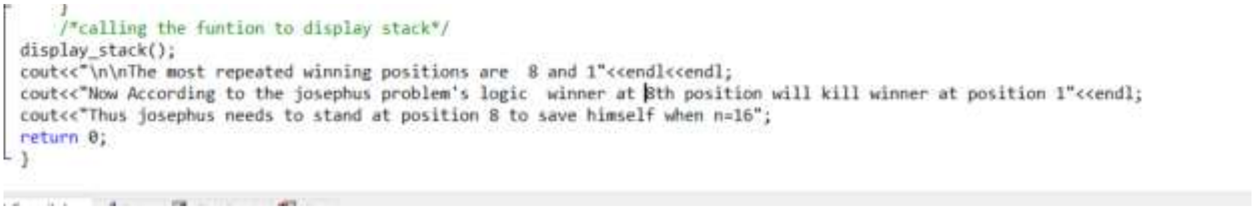
```
        else{
            ptr = (node*)malloc(sizeof(node));
            ptr->data = i+1;
            ptr_prev->next = ptr;
            ptr->prev = ptr_prev;
            ptr_prev = ptr;
        }
    }
    ptr->next = head;
    head->prev = ptr;
    return head;
}
 modified_josephus_problem(node *head, int m)
{
    node *ptr = head;
    if(ptr->next == head)
    return ptr->data;
    node *temp=ptr, *temp_prev = temp->prev;
    for(int i=1; i< m; i++)
    {
        temp_prev = temp;
        temp = temp->next;
    }
    node *new_head = temp->next;
    new_head->prev = temp_prev;
```



```
    new_head->prev = temp_prev;
    temp_prev->next = new_head;
    free(temp);
    return modified_josephus_problem(new_head, m);
}
struct Node* top = NULL;
void push_stack(int n)
 {
   struct Node* newnode = (struct Node*) malloc(sizeof(struct Node));
   newnode->data = n;
   newnode->next = top;
   top = newnode;
}

void display_stack()
 {
   struct Node* ptr;
   if(top==NULL)
   printf(" The Stack is empty");
      ptr = top;
      cout <<"Stack elements are :";
      while (ptr != NULL)
        {
      cout<<" "<<ptr->data;
      ptr=ptr->next;
```

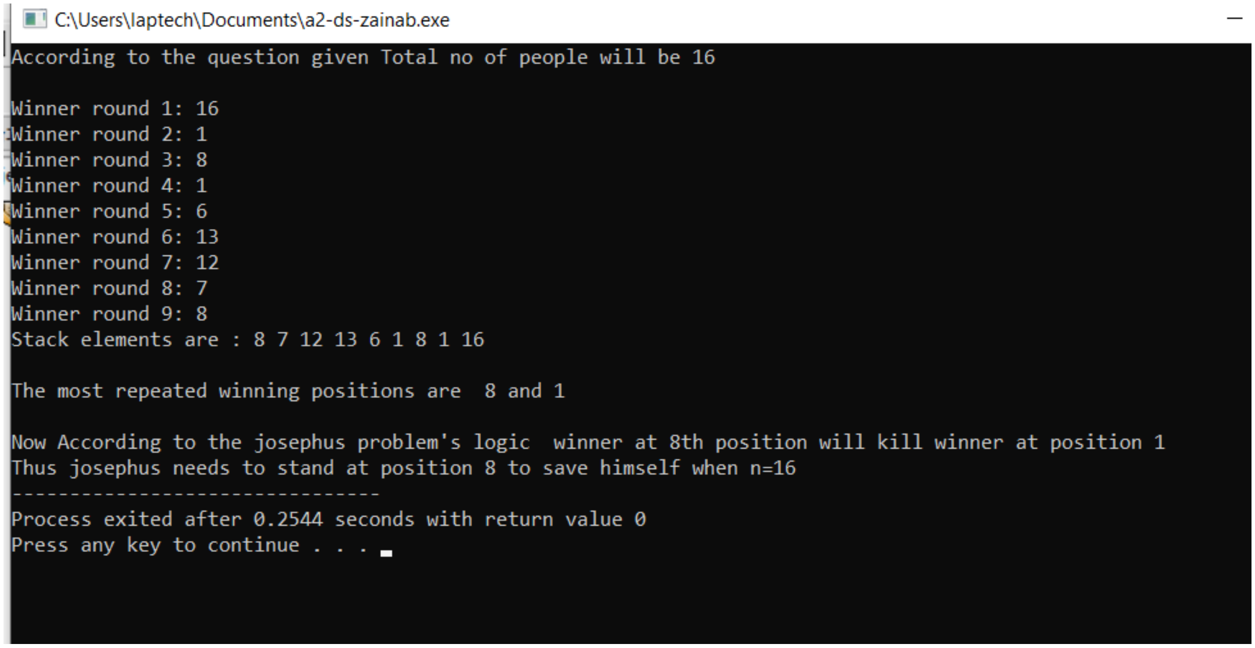


```
   }
int main()
{
    node *head = NULL;
    int n=16, m;
    cout<<"According to the question given Total no of people will be 16"<<endl<<endl;
    for(int l=0;l<9;l++)//leaders choosen till 9th position
    {
        int count=0;

        while (m!=9)
        {
        head = d_c_linkedlist(n);// doubly circular linked list
        m++;
        //position at which the person is killed is incremented each time a leader is choosed
        int winner;
        winner=modified_josephus_problem(head, m);
        push_stack(winner);
        count++;
        cout<<"Winner round "<< count<<": "<< winner<<endl;
        }
    }
    /*calling the funtion to display stack*/
```



```
    }
    /*calling the funtion to display stack*/
display_stack();
cout<<"\n\nThe most repeated winning positions are  8 and 1"<<endl<<endl;
cout<<"Now According to the josephus problem's logic  winner at 8th position will kill winner at position 1"<<endl;
cout<<"Thus josephus needs to stand at position 8 to save himself when n=16";
return 0;
}
```

Output:



```
According to the question given Total no of people will be 16

Winner round 1: 16
Winner round 2: 1
Winner round 3: 8
Winner round 4: 1
Winner round 5: 6
Winner round 6: 13
Winner round 7: 12
Winner round 8: 7
Winner round 9: 8
Stack elements are : 8 7 12 13 6 1 8 1 16

The most repeated winning positions are  8 and 1

Now According to the josephus problem's logic  winner at 8th position will kill winner at position 1
Thus josephus needs to stand at position 8 to save himself when n=16
--------------------------------
Process exited after 0.2544 seconds with return value 0
Press any key to continue . . .
```