DS LAB Week 6

Name: Adithya M SRN: PES1UG20CS621 SECTION: K

1) Joesphus Problem:

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct node
{
    int ele;
    struct node *link;
} node_t;

typedef struct queue
{
    node_t *front;
    node_t *rear;
} queue;

void init(queue *p)
{
    p->front = NULL;
    p->rear = NULL;
}

void enqueue(queue *p, int ele)
{
    node_t *temp = (node_t *)malloc(sizeof(node_t));
    temp->ele = ele;
    temp->link = temp;
    if (p->front == NULL)
    {
        p->front = temp;
        p->rear = temp;
    }
    else
    {
        p->rear->link = temp;
        temp->link = p->front;
        p->rear = temp;
    }
}

int front(queue *p)
{
    return p->front->ele;
}

void move(queue *p)
{
    p->front = p->front->link;
}
```

```c
void delete (queue *p)
{
    node_t *temp = p->front->link;
    if (temp == p->front)
    {
        return;
    }
    else
    {
        p->front->link = temp->link;
        free(temp);
    }
}

int main()
{
    queue q;
    init(&q);
    int n, k;
    printf("Enter n and k: ");
    scanf("%d %d", &n, &k);
    for (int i = 1; i <= n; i++)
        enqueue(&q, i);
    if (k == 1)
    {
        printf("Survivor: %d\n", n);
    }
    else
    {
        int count = 1;
        while (n)
        {
            if (count < k - 1)
            {
                move(&q);
                count++;
            }
            else
            {
                delete (&q);
                count = 0;
                n--;
            }
        }
        printf("Survivor: %d\n", front(&q));
    }
    return 0;
}
```

Output:

```
PS C:\Users\adith\Documents\C Programs> c
Enter n and k: 5 2
Survivor: 3
PS C:\Users\adith\Documents\C Programs\we
```

2) Queue operations:

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

#define MAX 20

typedef struct stack {
    int arr[MAX];
    int top;
}stack;

typedef struct queue {
    stack s;
}queue;

void init(stack *p) {
    p->top = -1;
}

int push(stack *p,int ele) {
    if(p->top == MAX-1) {
        return 0;
    }
    else {
        p->top++;
        p->arr[p->top] = ele;
    }
    return 1;
}

int pop(stack *p,int *ele) {
    if(p->top == -1) {
        return 0;
    }
    else {
        *ele = p->arr[p->top];
        p->top--;
    }
    return 1;
}

void initq(queue *p) {
    init(&p->s);
}

void enqueue(queue *p,int ele) {
    push(&p->s,ele);
}

void dequeue(queue *p) {
    stack temp; init(&temp);
    int ele;
```

```c
    int state;
    state = pop(&p->s,&ele);
    while(state){
        push(&temp,ele);
        state = pop(&p->s,&ele);
    }
    pop(&temp,&ele);
    state = pop(&temp,&ele);
    while(state){
        push(&p->s,ele);
        state = pop(&temp,&ele);
    }
}

void display(queue *q) {
    for(int i=0;i<=q->s.top;i++) {
        printf("%d ",q->s.arr[i]);
    }
    printf("\n");
}

int main() {
    queue q;
    initq(&q);
    int choice,ele;
    printf("1.Enqueue 2.Dequeue 3.Display 0:Exit\n");
    scanf("%d",&choice);
    do{
        switch(choice){
        case 1:
            printf("Enter the element :");
            scanf("%d",&ele);
            enqueue(&q,ele);
            break;
        case 2:
            dequeue(&q);
            break;
        case 3:
            display(&q);
            break;
        case 0:
            break;
        }
        printf("1.Enqueue 2.Dequeue 3.Display 0:Exit\n");
        scanf("%d",&choice);
    }while(choice);
    return 0;
}
```

Output:

```
PS C:\Users\adith\Documents\C Programs\week 6> cd "
1.Enqueue 2.Dequeue 3.Display 0:Exit
1
Enter the element :45
1.Enqueue 2.Dequeue 3.Display 0:Exit
1
Enter the element :876
1.Enqueue 2.Dequeue 3.Display 0:Exit
1
Enter the element :65
1.Enqueue 2.Dequeue 3.Display 0:Exit
3
45 876 65
1.Enqueue 2.Dequeue 3.Display 0:Exit
2
1.Enqueue 2.Dequeue 3.Display 0:Exit
3
876 65
1.Enqueue 2.Dequeue 3.Display 0:Exit
0
PS C:\Users\adith\Documents\C Programs\week 6>
```