

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

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

```
typedef struct node
```

```
{
```

```
    int data;
```

```
    struct node *next;
```

```
} node;
```

```
void push(node **head, int new_data)
```

```
{
```

```
    node *new_node = (node *)malloc(sizeof(node));
```

```
    new_node->data = new_data;
```

```
    new_node->next = NULL;
```

```
    if (*head == NULL)
```

```
    {
```

```
        *head = new_node;
```

```
    }
```

```
    else
```

```
    {
```

```
        node *temp = *head;
```

```
        while (temp->next != NULL)
```

```
        {
```

```

        temp = temp->next;
    }
    temp->next = new_node;
}
}

```

```

void pop(node **head)
{
    if (*head == NULL)
    {
        printf("Stack is empty\n");
    }
    else
    {
        node *temp = *head;
        node *prev = NULL;

        while (temp->next != NULL)
        {
            prev = temp;
            temp = temp->next;
        }

        if (prev == NULL)

```

```

    {
        *head = NULL;
    }

    else

    {
        prev->next = NULL;
    }

    printf("Popped element: %d\n", temp->data);
    free(temp);
}
}

```

```

void enqueue(node **front, int new_data)
{
    node *new_node = (node *)malloc(sizeof(node));
    new_node->data = new_data;
    new_node->next = NULL;

    if (*front == NULL)
    {
        *front = new_node;
    }

    else

```

```

{
    node *temp = *front;
    while (temp->next != NULL)
    {
        temp = temp->next;
    }
    temp->next = new_node;
}
}

```

```

void dequeue(node **front)
{
    if (*front == NULL)
    {
        printf("Queue is empty\n");
    }
    else
    {
        node *temp = *front;
        *front = temp->next;

        printf("Dequeued element: %d\n", temp->data);
        free(temp);
    }
}

```

```
}
```

```
void display(node *list)
```

```
{
```

```
    node *current = list;
```

```
    while (current != NULL)
```

```
    {
```

```
        printf("%d ", current->data);
```

```
        current = current->next;
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
int main()
```

```
{
```

```
    node *stack = NULL;
```

```
    node *queue = NULL;
```

```
    push(&stack, 1);
```

```
    push(&stack, 2);
```

```
    push(&stack, 3);
```

```
    printf("Stack: ");
```

```
    display(stack);
```

```

    pop(&stack);

    pop(&stack);

    pop(&stack);


    enqueue(&queue, 4);

    enqueue(&queue, 5);

    enqueue(&queue, 6);


    printf("Queue: ");

    display(queue);


    dequeue(&queue);

    dequeue(&queue);

    dequeue(&queue);


    return 0;
}

```

```

Stack: 1 2 3
Popped element: 3
Popped element: 2
Popped element: 1
Queue: 4 5 6
Dequeued element: 4
Dequeued element: 5
Dequeued element: 6
PS C:\Users\Admin\Desktop\1BM22CS272> 

```

