6 b)
WAP to implement single LinkList to simulate stack & Queue Operations.

```
#include <stdio.h>
#include <stdlib.h>
struct Node{
    int data;
    struct Node *next;
};
struct Node *top=NULL, *front=NULL, *rear=NULL;
struct Node* createNode (int value){
    struct Node *newNode=(struct Node*) malloc
            (sizeof (struct Node));
    if (!newNode) {
        Printf ("memory allocation failed!\n");
        exit(0);
    }
    newNode->data=value;
    newNode->next=NULL;
    return newNode;
}
void push(int value){
    struct Node * newNode=createNode(value);
    newNode->next=top;
    top=newNode;
    Printf ("%d pushed onto the stack\n",
                value);
}
```

```
Void pop(){
    if (top==NULL){
        Printf ("stack is empty.. Nothing to
                pop.\n");
        return;
    }
    struct Node *temp=top;
    Printf ("%d popped from the stack.\n",
            top->data);
    top=top->next;
    free (temp);
}
Void displayStack(){
    struct Node *temp=top;
    if (temp==NULL){
        Printf ("stack is empty.\n");
        return;
    }
    Printf ("stack (Top to Bottom): ");
    while (temp!=NULL){
        printf("%d ", temp->data);
        temp=temp->next;
    }
    printf("\n");
}
void enqueue(int value){
    struct Node *newNode=CreateNode(value);
    if (rear==NULL){
        front=rear=newNode;
    }else{
        rear->next=newNode;
        rear=newNode;
    }
    Printf("%d enqueued to the queue.\n", value);
}
```

```
void dequeue(){
    if (front == NULL){
        Printf ("Queue is emtpy. Nothing to
        dequeue.\n");
        return;
    }
    struct Node *temp = front;
    printf ("%d deleted from the queue.\n",
            front->data);
    front= front->next;
    if (front== NULL){
            rear =NULL
    }
    free (temp);
}
void displayQueue(){
    Struct Node *temp =front;
    if (temp== NULL){
        Printf ("Queue is emtty:\n");
        return;
    }
    Printf ("Queue(front to Rear): ");
    while (temp!=NULL){
        Printf ("%d ", temp->data);
        temp = temp->next;
    }
    Printf ("\n");
}
```

```
int main(){
    int choice, value, ch;
    while (1){
        Printf("\n -- Singly Linked List simulation --\n");
        Printf ("1. Stack operations\n");
        Printf ("2. Queue Operation\n");
        Printf ("3. Exit\n");
        Printf ("Enter your choice:");
        scanf ("%d", &choice);
        switch (choice){
            case1:
                while(1){
                    printf ("\n --stackmenu --\n");
                    Printf ("1. Push\n");
                    printf("2. pop\n");
                    Printf("3. display stack\n");
                    printf("4. Back to main menu\n");
                    Printf("enter your choice: ");
                    scanf("%d", &ch);
                    switch(ch){
                        case 1:
                            Printf ("enter value to enqueue: ");
                            scanf("%d", &value);
                            enqueue(value);
                            break;
                        case 2:
                            dequeue(); pop();
                            break;
                        case 3:
                            displayQueue();
                            break;
```

```
            case 4:
                goto main_meny;
            default:
                Printf("Invalid choice.\n");
        }
    }
    break;

case 2:
  while(1){
    printf("\n--Queue Menu--\n");
    Printf("1.Enqueue\n");
    Printf("2.Dequeue\n");
    Printf("3.Display Queue\n");
    printf("4.Back to main menu\n");
    Printf("Enter your choice:");
    Scanf("%d",&ch);
    Switch(ch){
      case 1:
        Printf("Enter value to enqueue:");
        Scanf("%d",&value);
        enqueue(value);
        break;
      case 2:
        dequeue();
        break;
      case 3:
        displayQueue();
        break;
```

```
      case 4:
        goto main_menu;
      default:
        Printf("Invalid choice.\n");
    }
  }
  break;

case 3:
    Printf("Exiting program.\n");
    exit(0);
default:
    Printf("Invalid choice. Try again.\n");
}
main_menu;;
}
return 0;
```