6b) WAP to implement singly link list to simulate stack and queue operations.

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

Struct node {
        int data;
        Struct node *Next;
  };

Struct node * top = NULL;
Struct node * front = NULL;
Struct node * ~~f~~ rear = NULL;
```

8

```
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 * new node = 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 empty. nothing to
                 dequeue.\n");
   return;
  }.
  struct Node * temp = front;
  printf ("%d dequeued 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 empty .\n");
    return;
  }
   printf (" Queue [front to Rear ]:");
   while (temp != NULL)
    {
       printf (" %d ", temp->data);
       temp = temp-> ~~data~~ 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 operations\n");
printf ("3. Exit \n");
printf ("Enter your choice:");
scanf ("%d", &choice);

switch (choice)
{
  case1 :
        while(1){
            printf ("\n --- stack menu ---\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){
        case1:
              printf ("Enter value to
                          push:");
              scanf ("%d", &value);
              push (value);
               break;

        case2:
               pop ()
               break;
        case3:
             display stack();
             break;
        case4:
               goto main_menu;
        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.
                          Try again.\n");
        }
}
```

```
break;

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

output:

```
-- singly linked list simulation --
1 stack operation
2 queue operation
3  exit.
Enter your choice: 1

--  stack --- menu
1. push
2. pop
3  Display stack
4. Back to main menu
Enter your choice : 3
 Stack is empty.

--- stack menu -----
1. push
2. pop
```