B.M.S. COLLEGE OF ENGINEERING

Bull Temple Road, Basavanagudi, Bengaluru-590019, Karnataka.



Data Structures using C Lab

(23CS3PCDST)

Submitted by

Sinchana Hemanth (1BM23CS330)

in partial fulfillment for the award of the degree of BACHELOR OF ENGINEERING in COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019 Sep-2024 to Jan-2025

B.M.S. College of Engineering,

Bull Temple Road, Bangalore 560019 (Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled "Data Structures using C Lab (23CS3PCDST) carried out by Sinchana Hemanth (1BM23CS330), who is a bonafide student of B.M.S. College of Engineering. It is in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of Data Structures using C Lab (23CS3PCDST) work prescribed for the said degree.

Prasad GR Assistant Professor Department of CSE, BMSCE Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

2

Index

Sl. No.	Date	Experiment Title	Page No.
1	01/10/24	Stack Implementation using arrays	4
2	07/10/24	Parenthesized infix arithmetic expression to postfix expression	12
3(a)	14/10/24	Insertion, deletion and display in queues using arrays	21
3(b)	21/10/24	Insertion, deletion and display in circular queues using arrays	34
4	28/10/24	Creation, insertion and display of nodes in a linked list	43
5	10/11/24	Deletion operations in a linked list	49
6(a)	01/12/24	Sort, reverse and concatenate linked lists	61
6(b)	01/12/24	Stack and queue implementation using linked lists	69
7	08/12/24	Insert, delete and display nodes in doubly linked list	80
8	22/12/24	Construct and traverse a binary search tree	92
9(a)	22/12/24	Traverse a graph using BFS method	101
9(b)	22/12/24	Traverse a graph using DFS method	106

Github Link: https://github.com/SinchanaHemanth/1BM23CS330-SinchanaHemanth.git 3

Program 1

Write a program to simulate the working of stack using an array with the following:

- a) Push
- b) Pop
- c) Display

The program should print appropriate messages for stack overflow, stack underflow.

Implement stack and cops us	
#include (stalled. L>	Profession
#include < statio. h>	
int top, size;	1 100 x - x 903
The section of the se	and the state of t
youd init()	1
3	7.4
printe ("Enter size of stack In")	1 constant
scare (" 6 1. d", fisige);	and a second
top = -1;	to the second of
a contract of the contract of	ort court
bod isEmpty()	ACA ANTONIO
1	N NEW PLANT
return top==-1;	and the state of
3	a 1-24a) - 2 9ay
	- The second second
bool is full()	
C	Torres O
neturn top==sige-1;	
	THE / East 1 128
Company of the control of the contro	and water
void push (int ann [7, int n)	DOLLAR TO THE PARTY OF THE PART
(Harris Harris Harris	Course Carl
if (!IsFuell)	and all the state of
C	TOTAL STATE
top++;	and the start of the
our [top] = x;	and the same to skill a sta
printe ("Aushed V.d to stack	In", se);
9	Section States of the State of States

	Progo No. Dete
	(all stores
	private ("Overflow In");
	3
	3 comments in the day of the state of the st
	int poplint and [7)
	C ROSPOSITO DASS
	iy (isEmpay())
	1 Marie Character Tay Vent
	printe ("Undergeau in");
	neturn O;
	3
	else (Herman 13 - Jan 1 mg
	1
	int temp = and [top];
	top;
	return temp;
	3
	3
	int peck (int arr [])
	E peck (but ask ())
	print("Stack is empty In");
) is	else that the state of
- Chiese	1
	gueturn arr [top];
	3
K Kale	St. 2 Causia. Ca.
100	int main()
5 1 1 2	· Links and the Control of the Contr
	Section 1 - Francisco de la contraction de la co

	Paga No. Data
	init();
	int and Isige?;
	AND THE PROPERTY AND ADDRESS OF THE PARTY AND
	for (int l=0; l <size; i++)<="" td=""></size;>
	Cart and the cart
	int element;
	printe (" Eroter element 1.d: , i+1);
	sant (" ".d", felement);
	push (arr, clement);
	3 miles of the state of the sta
	for (int i=0; ixsige; i++)
	ξ
	print ("Popped: 7.d In", pop (ass));
Seen	neturn 0;
	9
Carlon	
output	Enter size of stack:
	5
	Enter element 1: 6 (1) (1) (1)
	Pushed & to slack
T. C. N.	Enter element 2: 7
	Pushed 7 to stack
Charles Dr.	Enter element 3: 8
17.10	Pushed 8 to stack
100	Ereca clement 4:9
	Pushed 9 to stack
Wall In	Ender element 5: 10
	Pashed 10 to stack
Water and the second	Popped: 10
33	Papped: 9
	Popped: 8 Popped: 6

```
c stacks.c > 0 main()
      #include <stdio.h>
      #define MAX 5
      int stack[MAX];
      int top = -1;
      void push(int value) {
          if (top == MAX - 1) {
               printf("Stack Overflow! Cannot push %d\n", value);
              top++;
              stack[top] - value;
              printf("%d pushed into the stack.\n", value);
      void pop() {
          if (top == -1) {
               printf("Stack Underflow! Cannot pop.\n");
              printf("%d popped from the stack.\n", stack[top]);
              top--;
      void display() {
          if (top -- -1) {
               printf("Stack is empty.\n");
              printf("Stack elements: ");
               for (int i = 0; i \leftarrow top; i++) {
                   printf("%d ", stack[i]);
               printf("\n");
```

```
int main() {
         int choice, value;
         while (1) {
             printf("\nStack Operations:\n");
             printf("1. Push\n");
             printf("2. Pop\n");
             printf("3. Display\n");
             printf("4. Exit\n");
             printf("Enter your choice: ");
             scanf("%d", &choice);
             switch (choice) {
                 case 1:
                     printf("Enter the value to push: ");
                     scanf("%d", &value);
                     push(value);
                     break;
                 case 2:
                     pop();
                     break;
                     display();
                     break;
                 case 4:
                     printf("Exiting...\n");
                     return 0;
                 default:
                     printf("Invalid choice! Please try again.\n");
69
```

8 **OUTPUT:**

```
Stack Operations:
1. Push
2. Pop
Display
4. Exit
Enter your choice: 1
Enter the value to push: 5
5 pushed into the stack.
Stack Operations:

    Push

2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to push: 10
10 pushed into the stack.
Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to push: 15
15 pushed into the stack.
Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to push: 20
20 pushed into the stack.
```

```
PROBLEMS
          OUTPUT DEBUG CONSOLE
                                   TERMINAL
                                               PORTS
Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to push: 25
25 pushed into the stack.
Stack Operations:

    Push

2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to push: 30
Stack Overflow! Cannot push 30
Stack Operations:
1. Push
2. Pop
Display
4. Exit
Enter your choice: 3
Stack elements: 5 10 15 20 25
Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
25 popped from the stack.
```

```
Stack Operations:
1. Push
2. Pop
Display
4. Exit
Enter your choice: 2
20 popped from the stack.
Stack Operations:
1. Push
2. Pop
Display
4. Exit
Enter your choice: 2
15 popped from the stack.
Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
10 popped from the stack.
Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
5 popped from the stack.
```

```
Stack Operations:

1. Push

2. Pop

3. Display

4. Exit
Enter your choice: 2
Stack Underflow! Cannot pop.

Stack Operations:

1. Push

2. Pop

3. Display

4. Exit
Enter your choice: 4
Exiting...
PS C:\Users\TOSHIBA\Documents\UiPath\TRIAL\dsa>

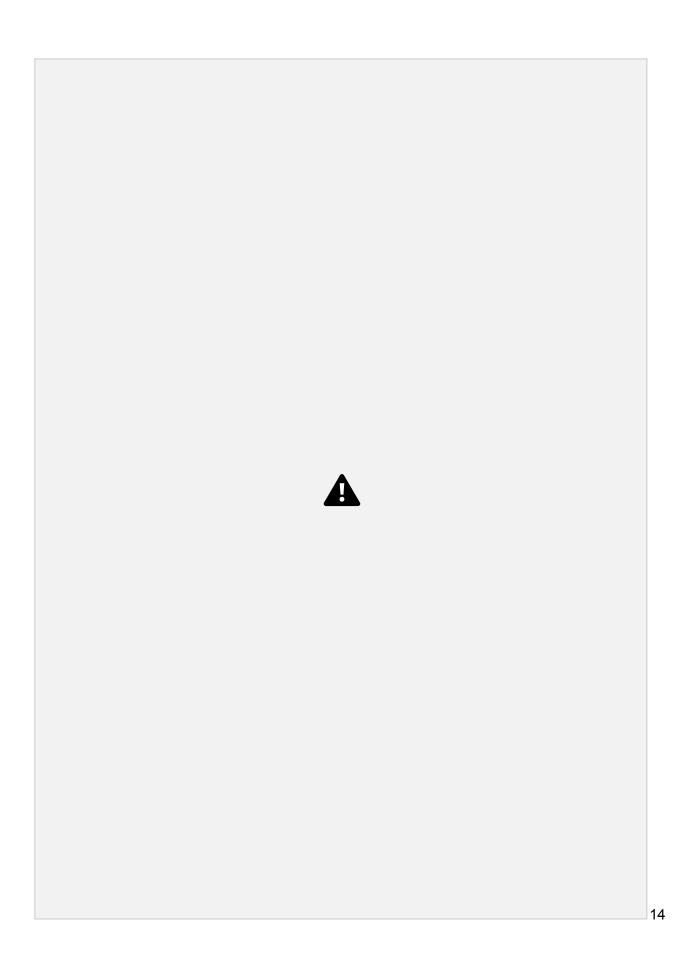
[]
```

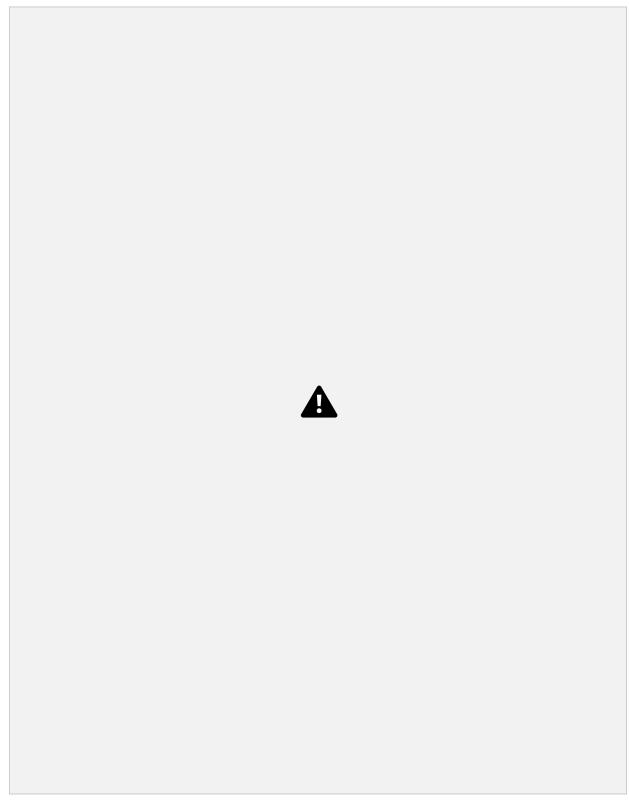
Program 2

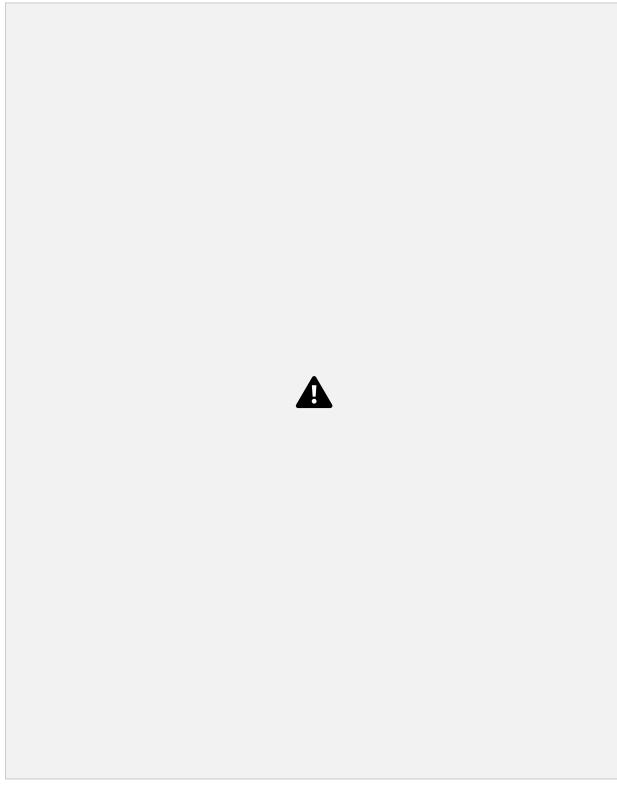
expressions. The expression consists of single character operands and the binary operators are + (plus), - (minus), * (multiply), / (divide) and $^$ (exponential).

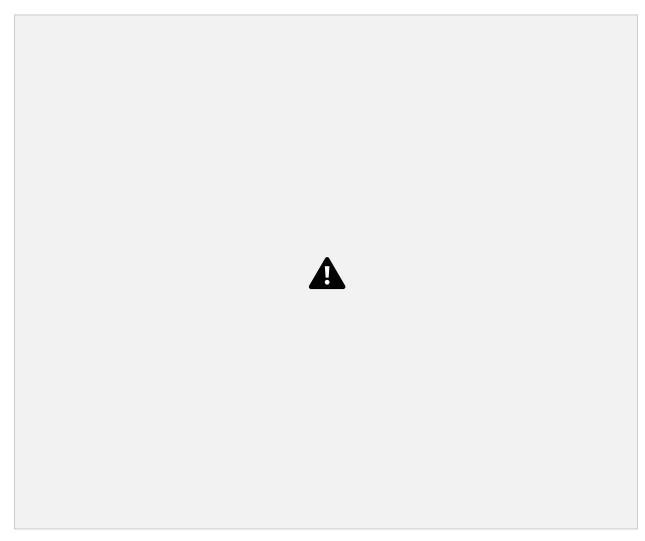
	Pege No. Gene
	Inb program-2
	#indude Koldio. hs
	# include { come bostdlib h >
	# include Latype. hs
	# include String. hs
	# degine MAX 100
	- /s/2 - 1
	typede struct
-	r state of the sta
	int too:
	char items [max];
	3 Stack;
	1.2
	yad initotack (stack *s)
	E Transfer Comments
	6-> top = -1;
	3 Company (Tell asserts noted asserts to a large 1 services 1
	A Continuous A
	int is Emptyl Stack *5)
	5
	outurn 6->top==-1;
	The second secon
	n Internacion
	vad push (stack *s, char item)
	1 Charter & State of Land to the State of State of the St
+	y (6 > top < MAX -1)
+	(
+	5-> items [++ (5->top)] = item;
-	3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
-	Upe
	printe ("stack overflow In");

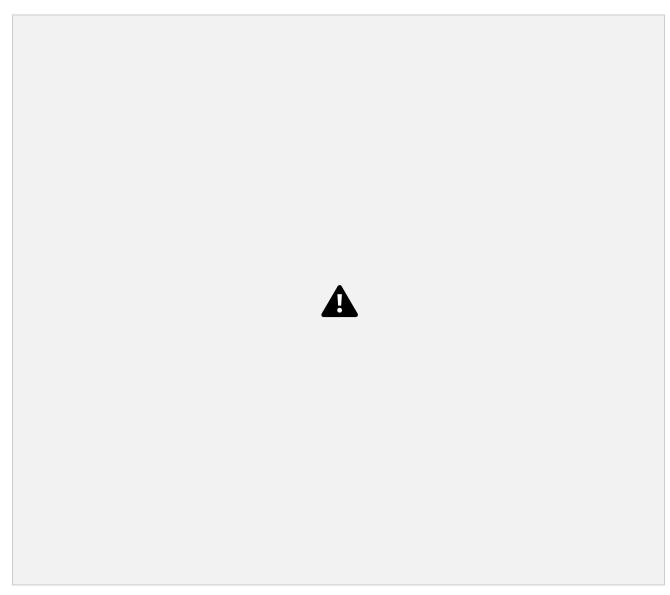
-	char & pgp (Stack *5)	
	2 - Marin James Barry Bridge Barry B	
	y (!isEmpty(s))	
	(
	neturn s > items [(s > top) -];	
	3	
	else	
	[printe ("Stack underglow In");	
	return'10; 3	
	3. Secretary	
		Į,
	alon mul (Graduta)	
	char peek (Stack +s)	_
	: (-1/25	
	ig lo! (SEmpty (S))	_
	Contract and Colored Total Contract and Cont	
	return s → items [s > tap];	_
	(1 - 1207 > California)	-
	else	_
	2-2 month for to san H = non i	_
	greturn (10);	_
	3	Č.
)	P

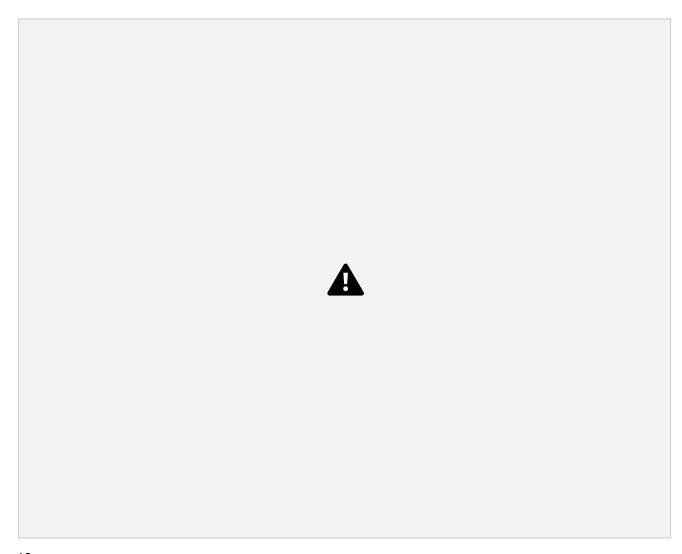


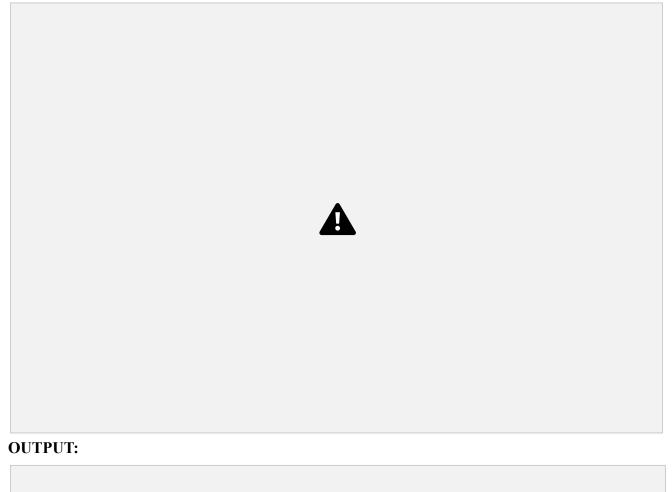










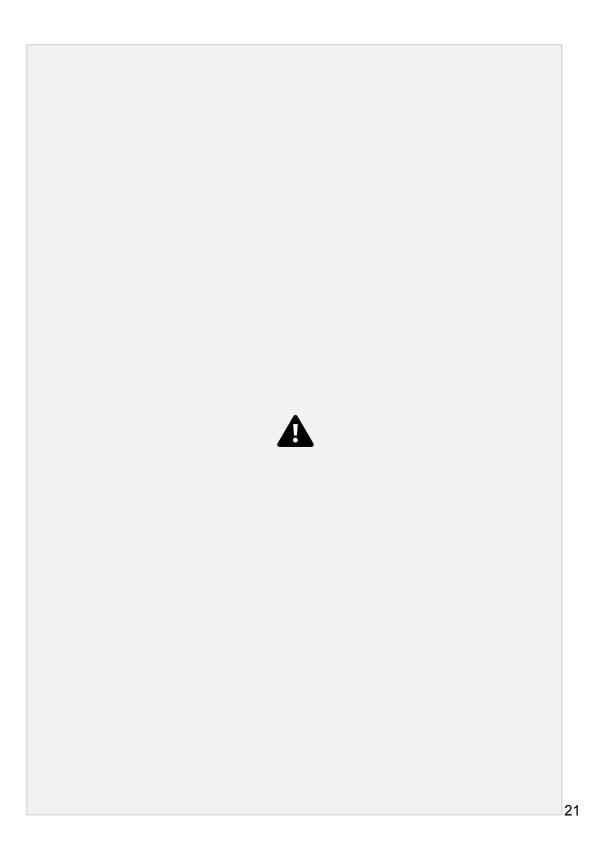


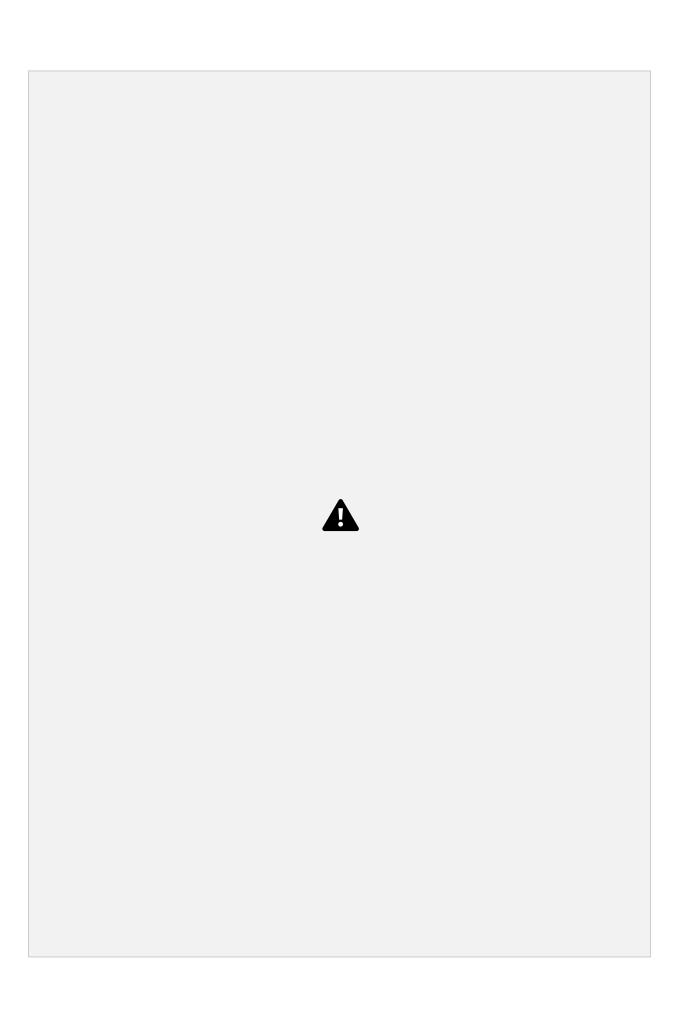


20

Program 3

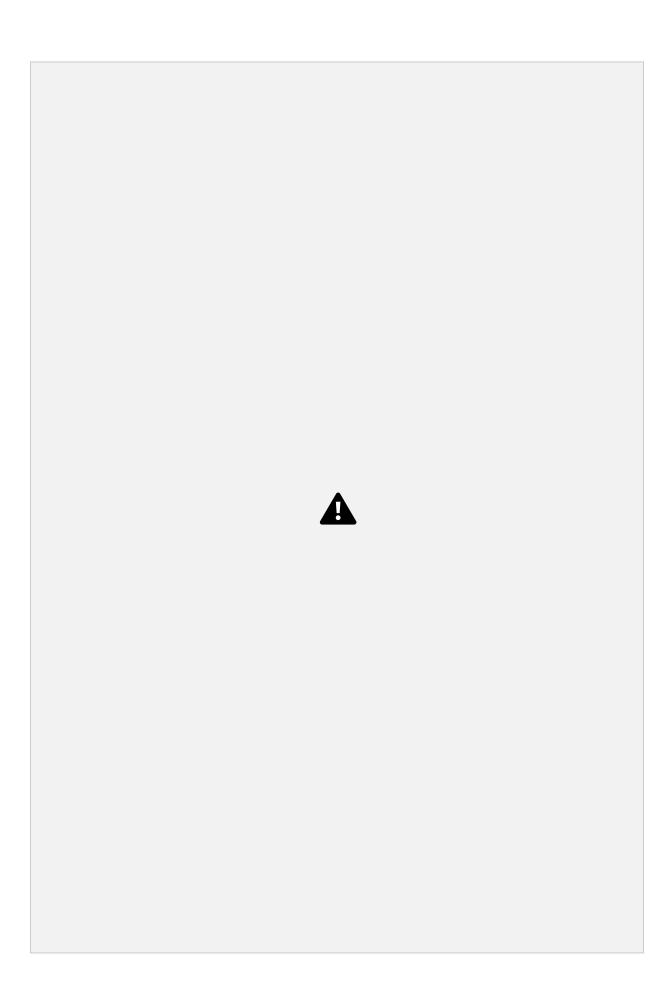
(a) Write a C program to simulate the working of a queue of integers using an array. Provide the following operations: insert, delete, display. The program should print appropriate messages for queue empty and queue overflow conditions.



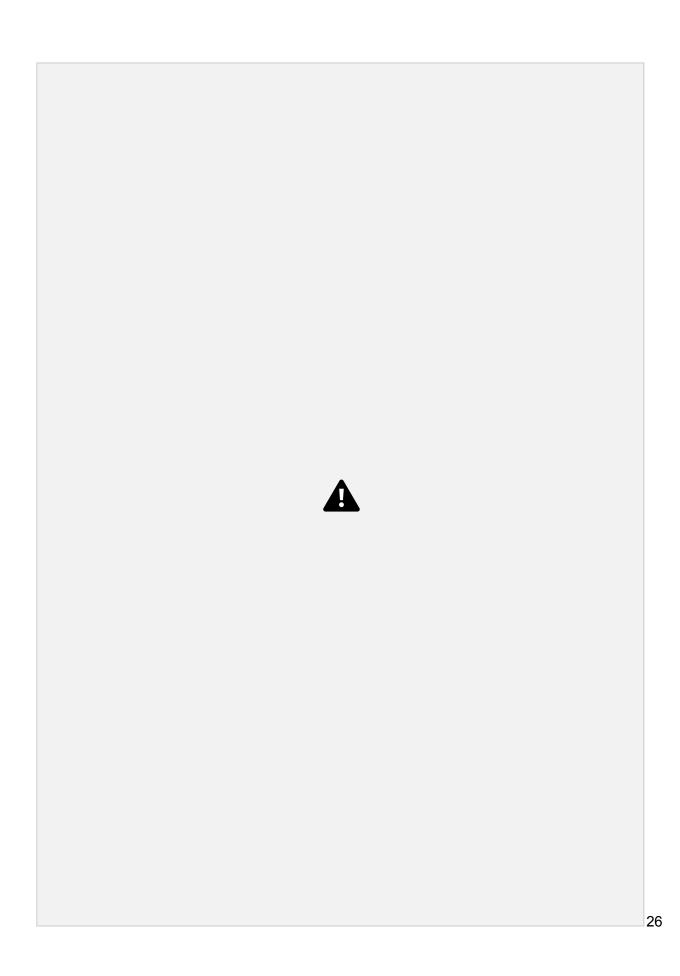


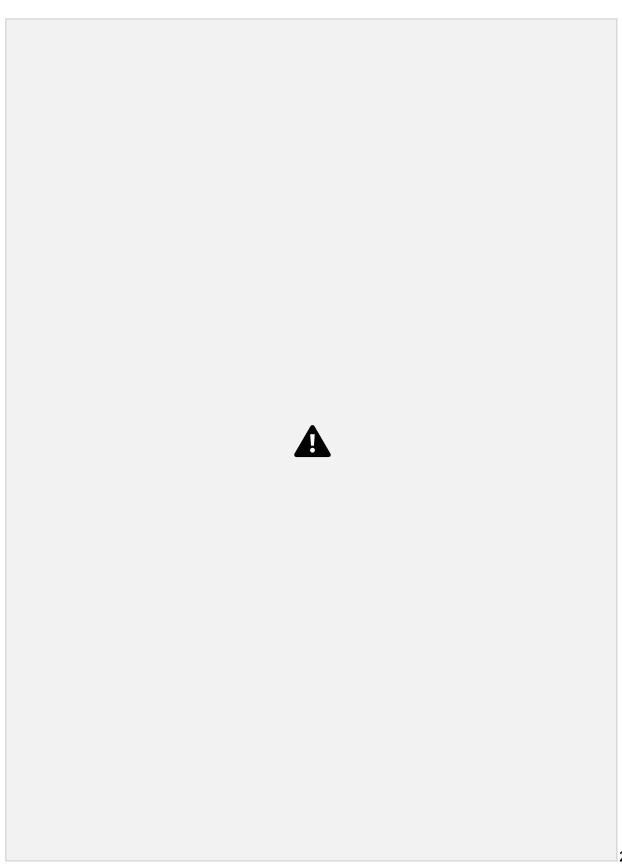






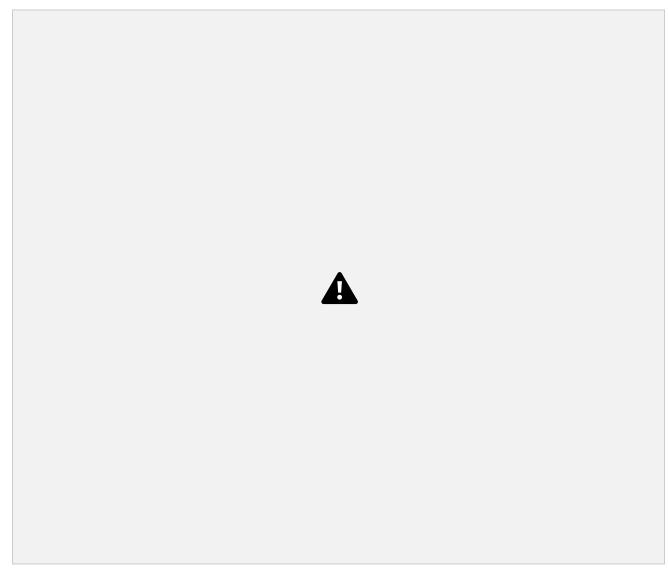






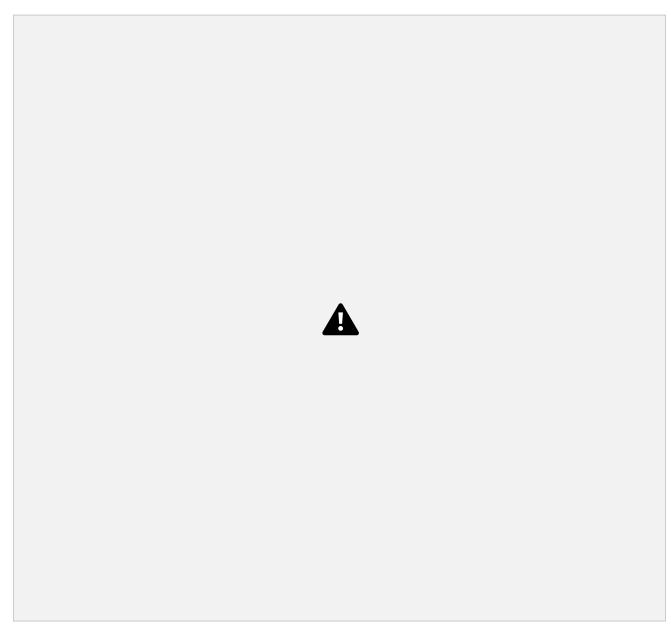


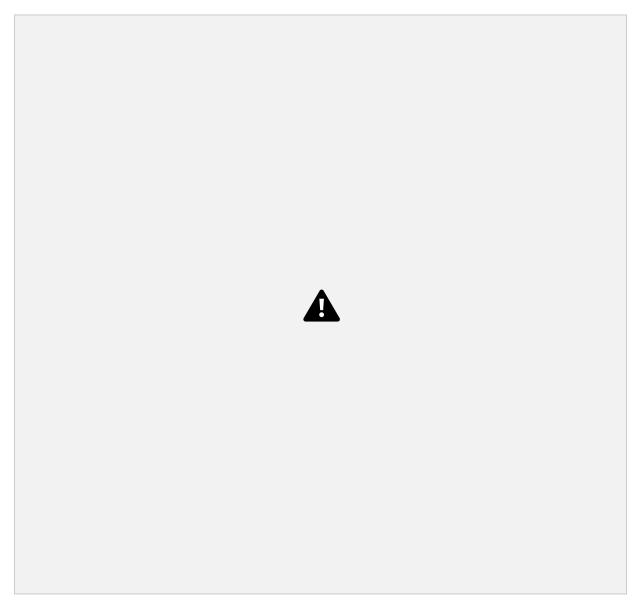


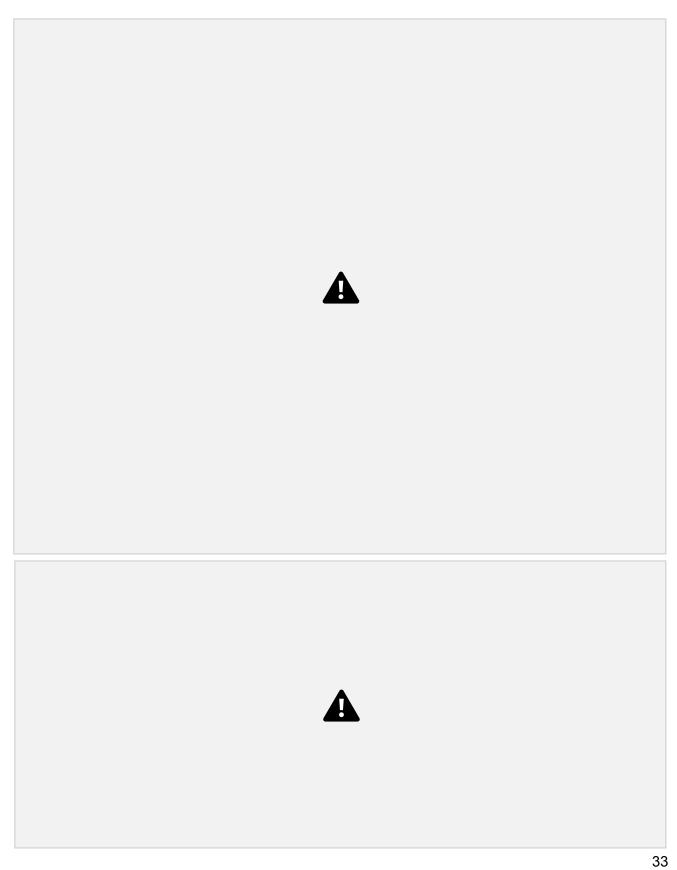


30

OUTPUT:

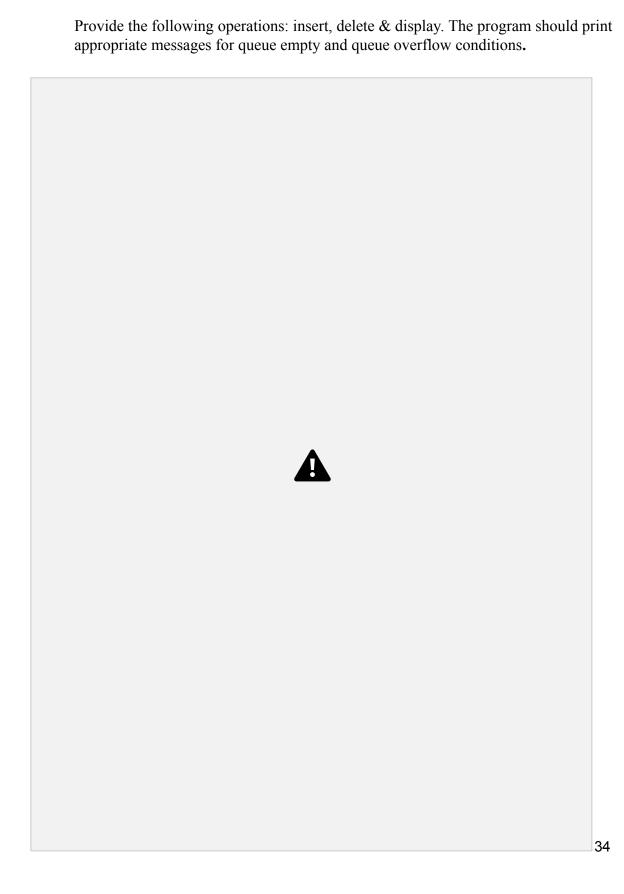


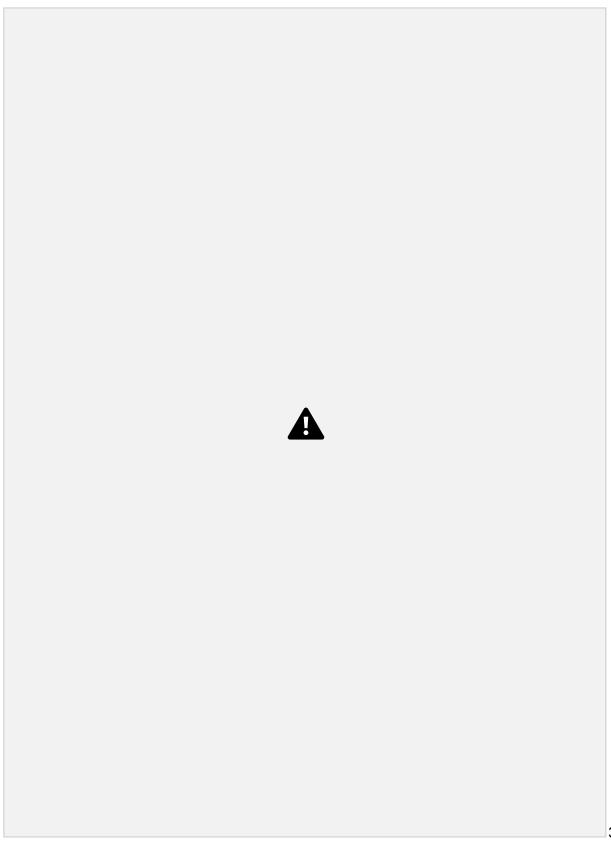


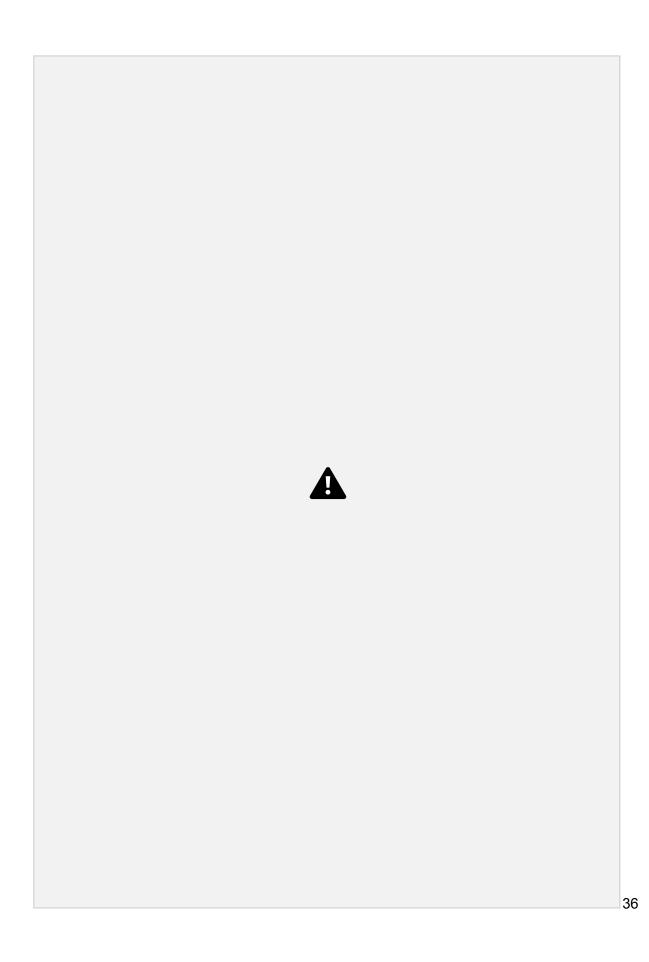


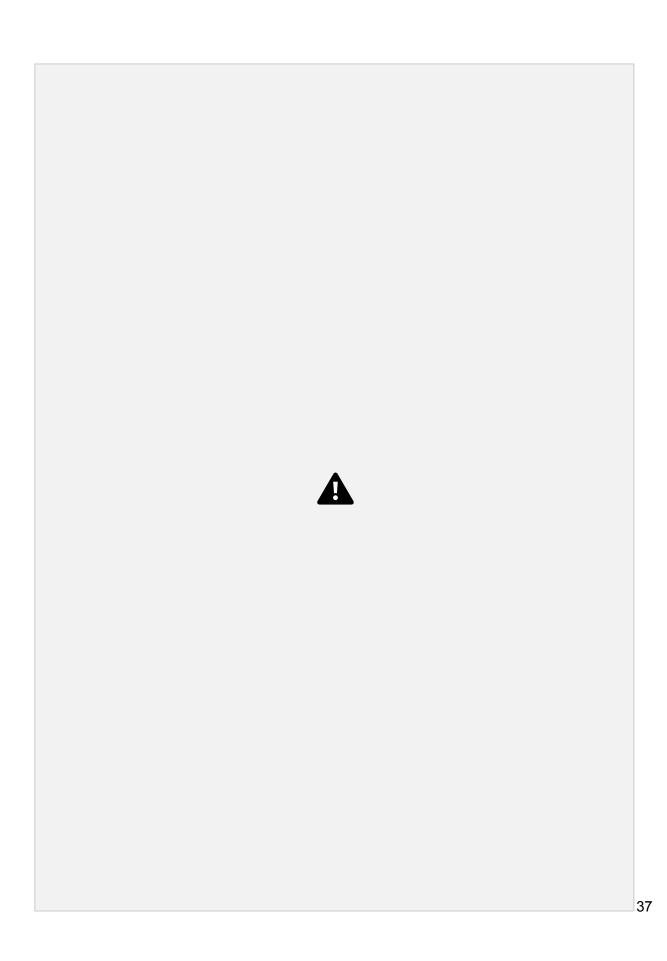
Program 3

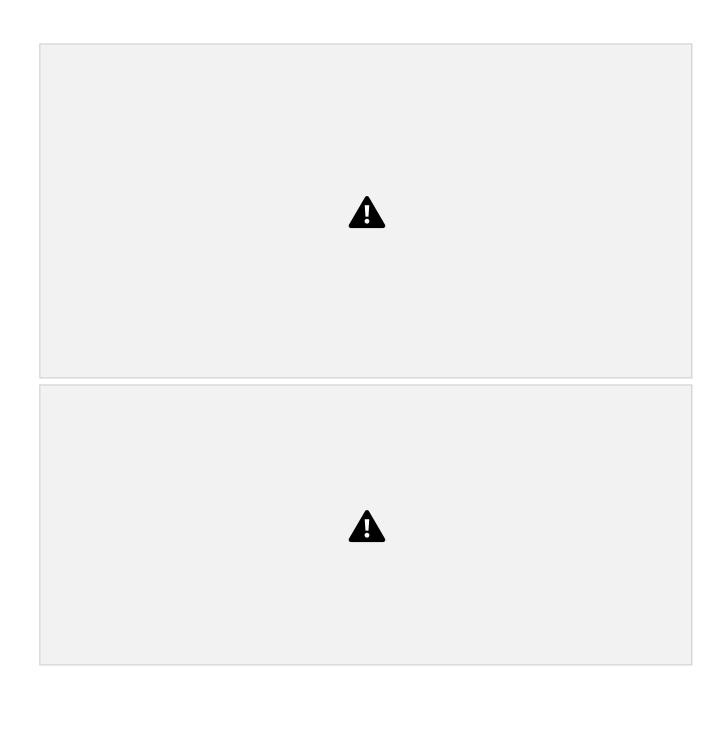
(b) Write a program in c to simulate the working of a circular queue of integers using an array.



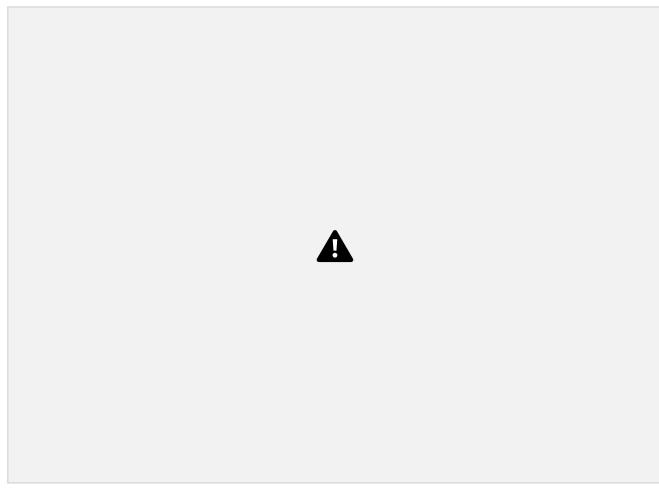




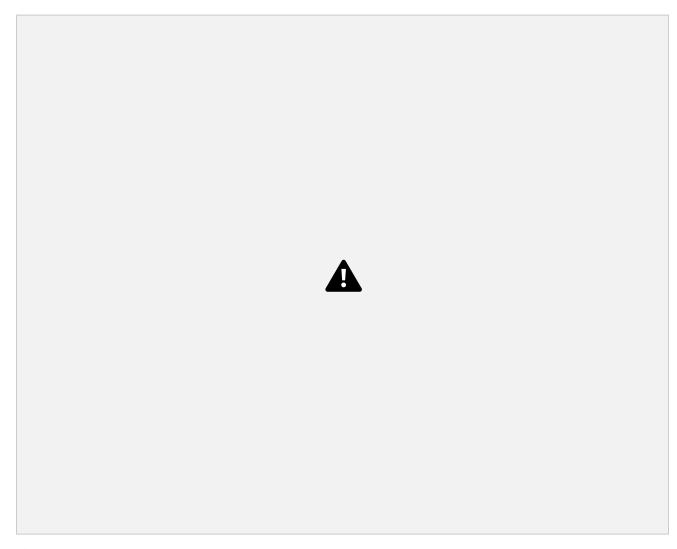


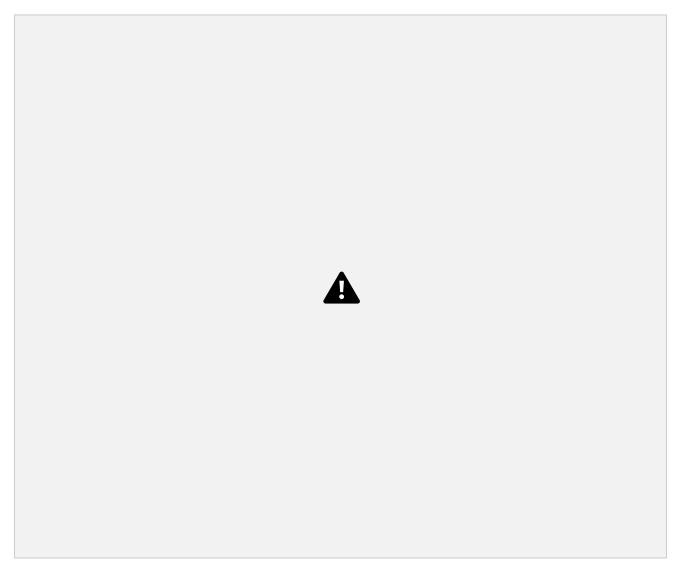


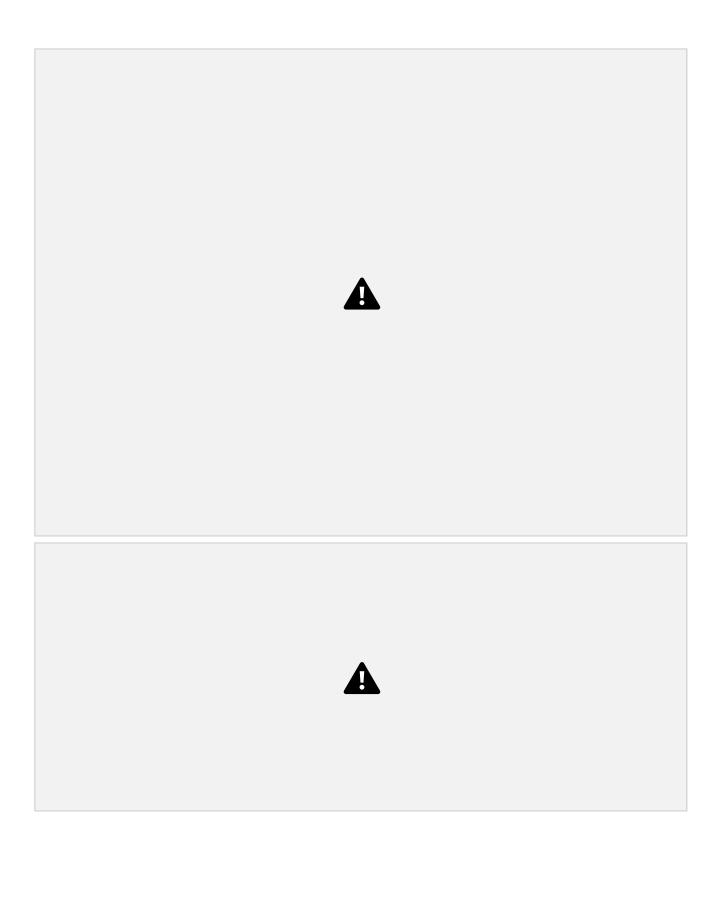




OUTPUT:



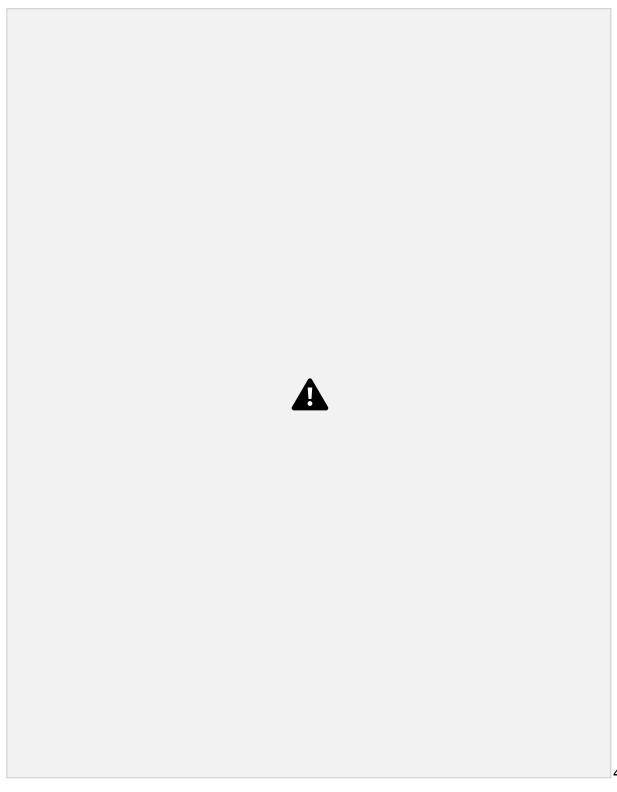


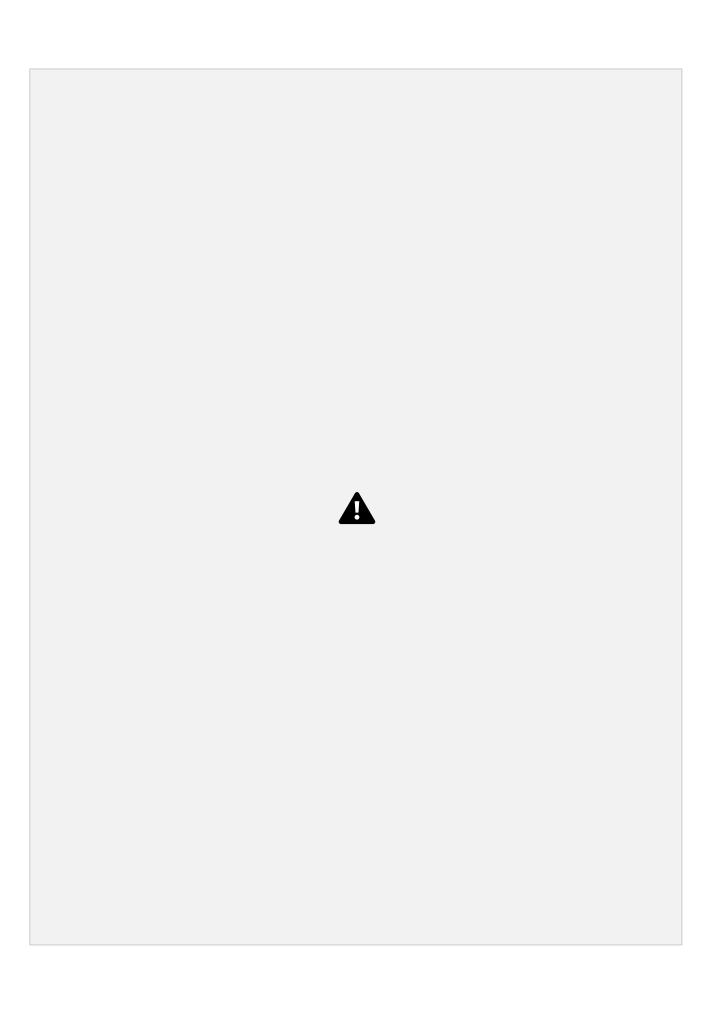


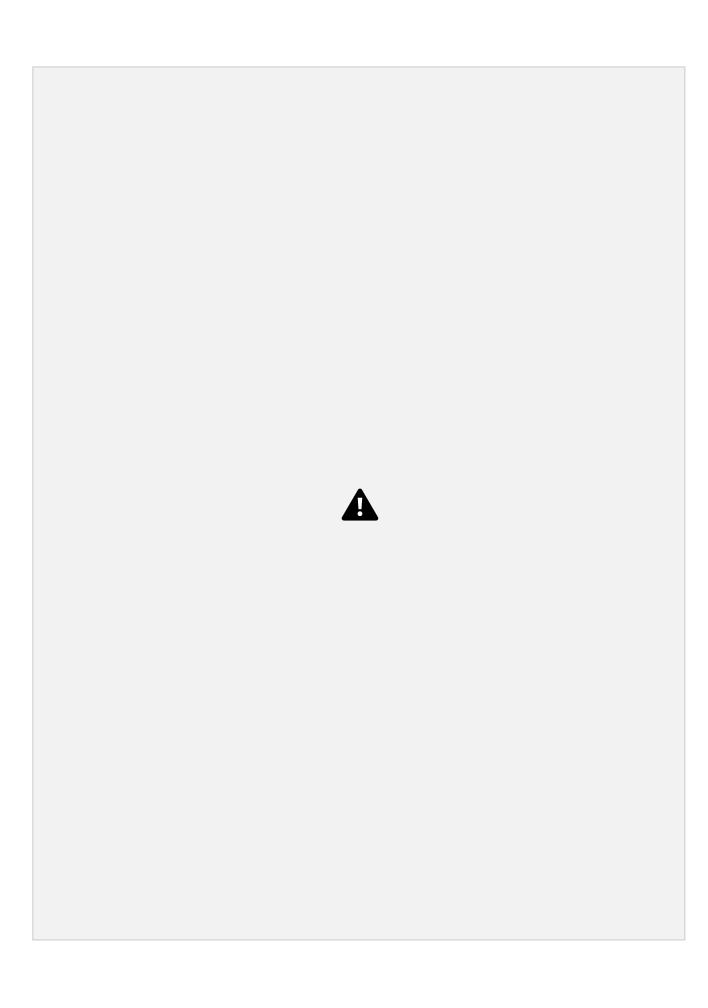
Program 4

WAP in c to Implement Singly Linked List with following operations:

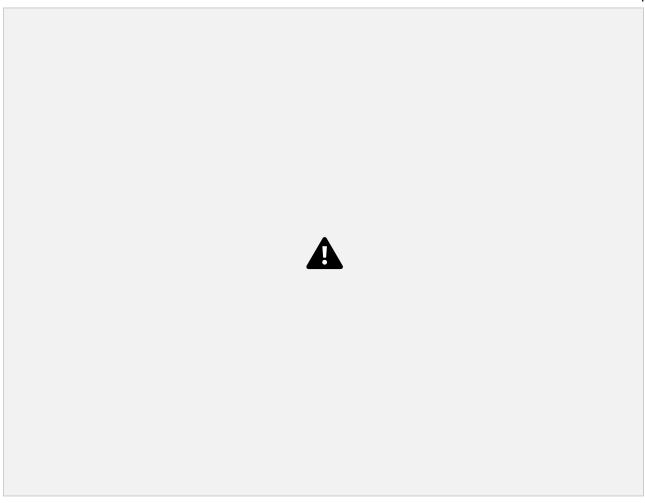
- (a) Create a linked list
- (b) Insertion of a node at first position and at end of list
- (c) Display the contents of the linked list.

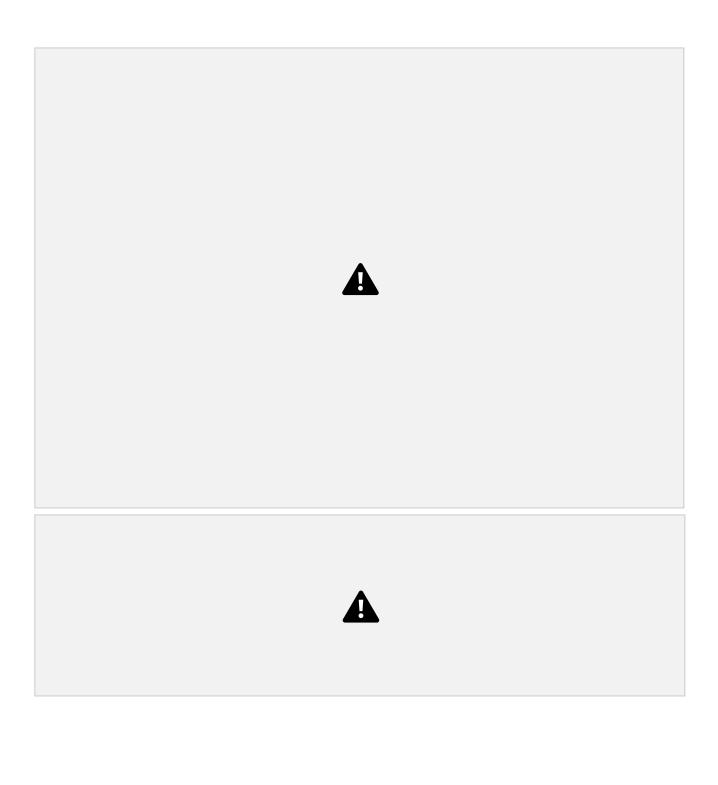
















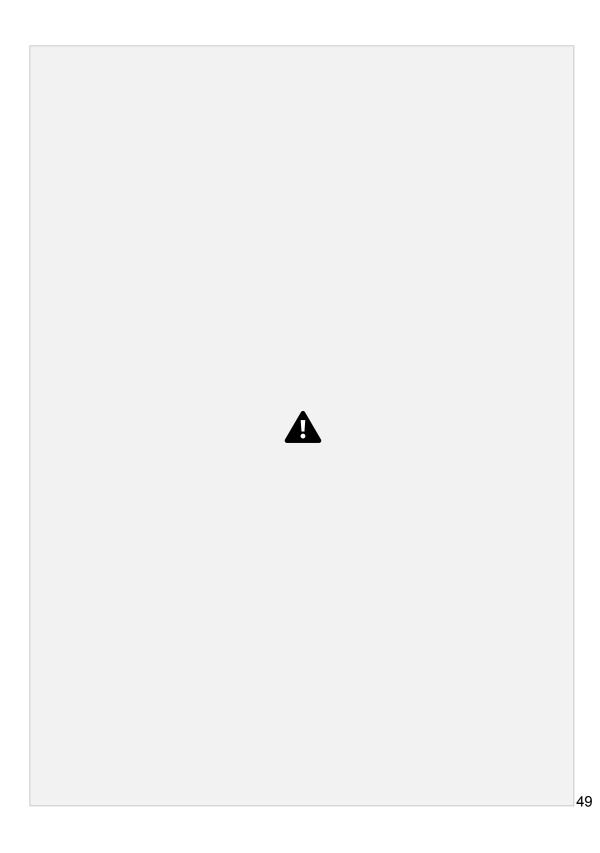


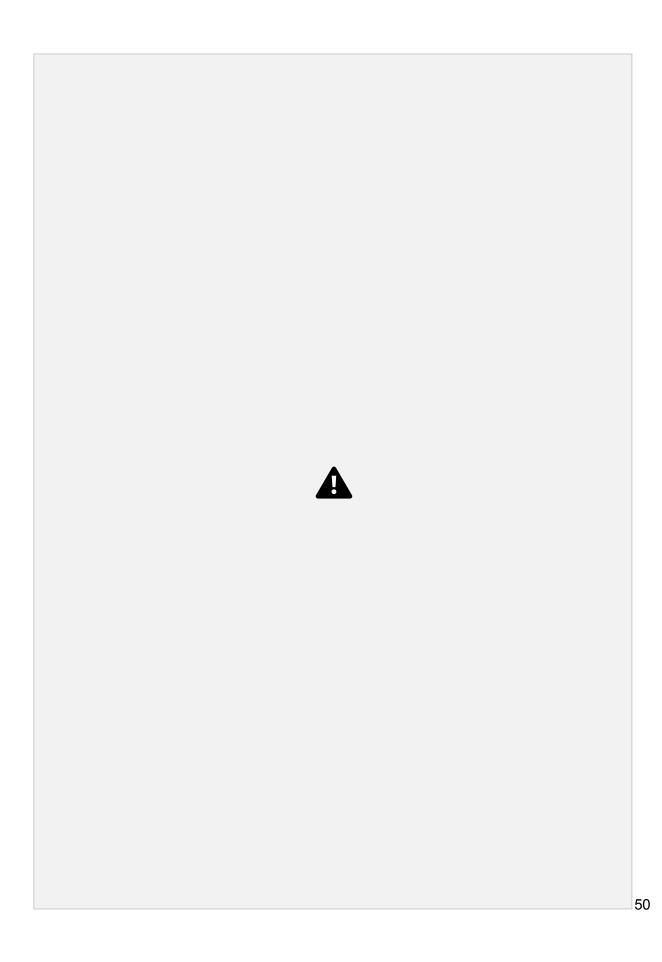
Program 5

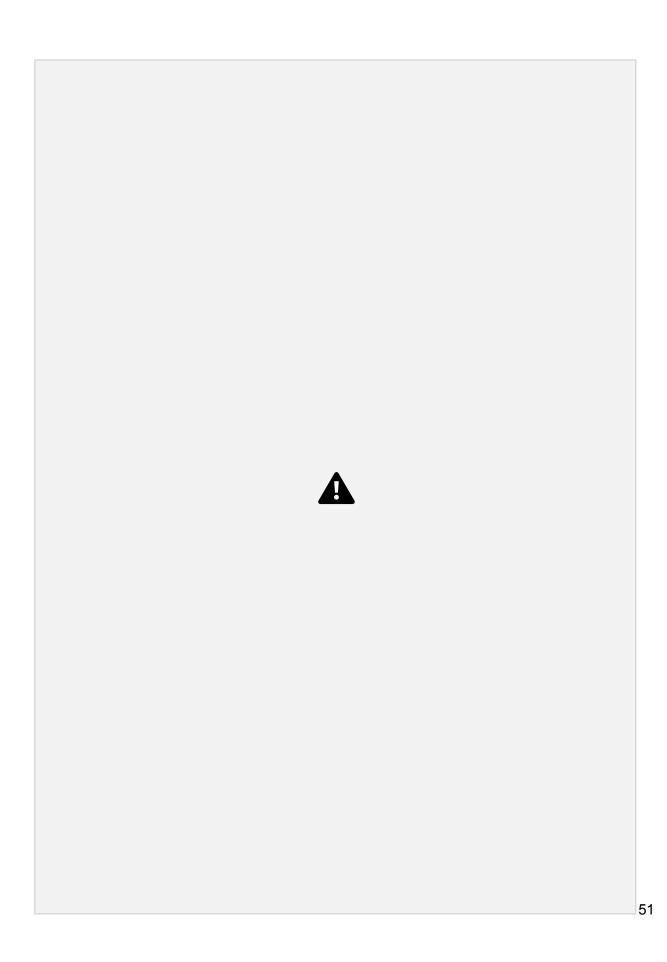
WAP in C to Implement Singly Linked List with following operations:

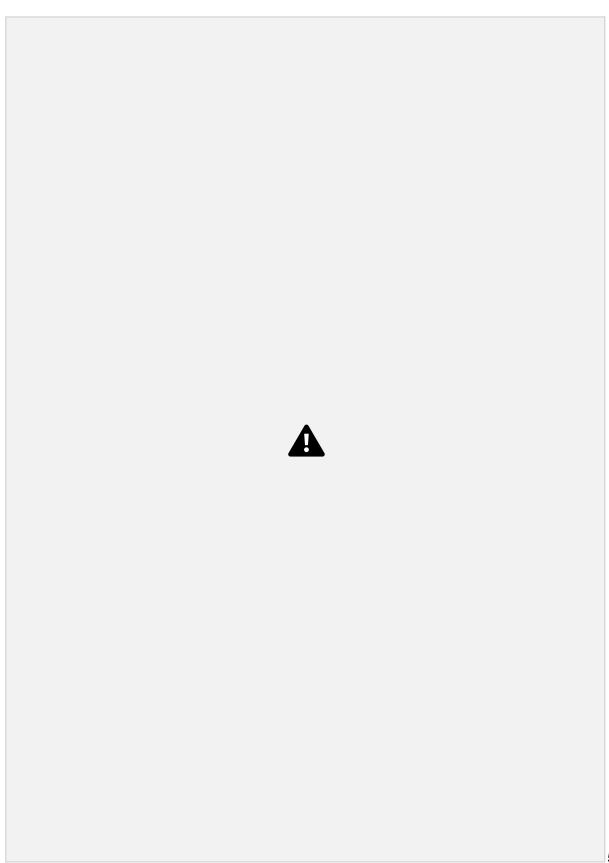
- (a) Create a linked list.
- (b) Deletion of the first element, specified element and last element in the list.
- (c) Display the contents of the linked list.

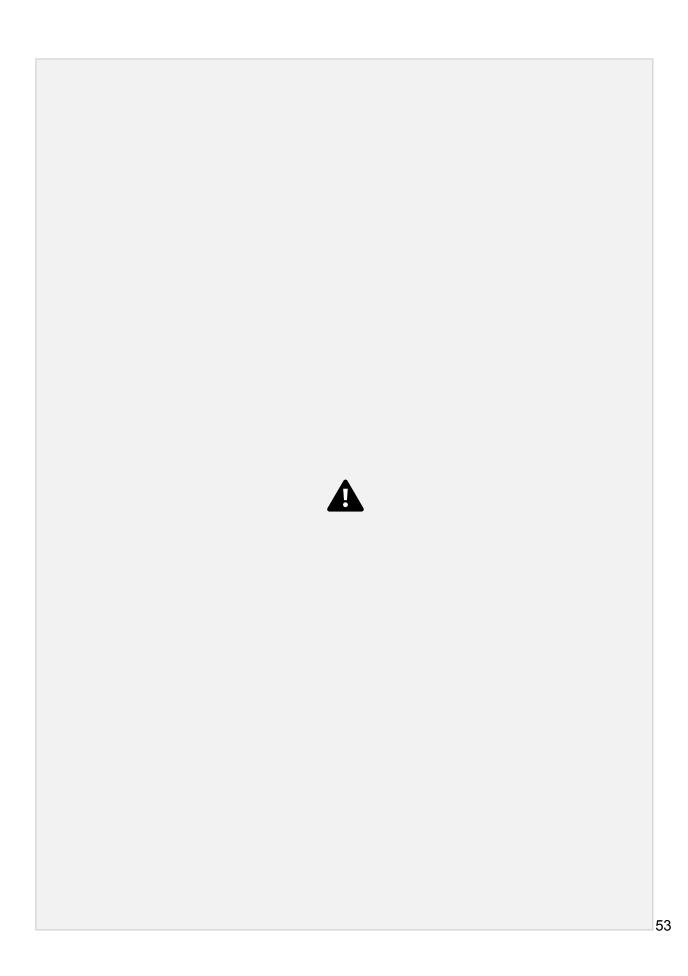
48

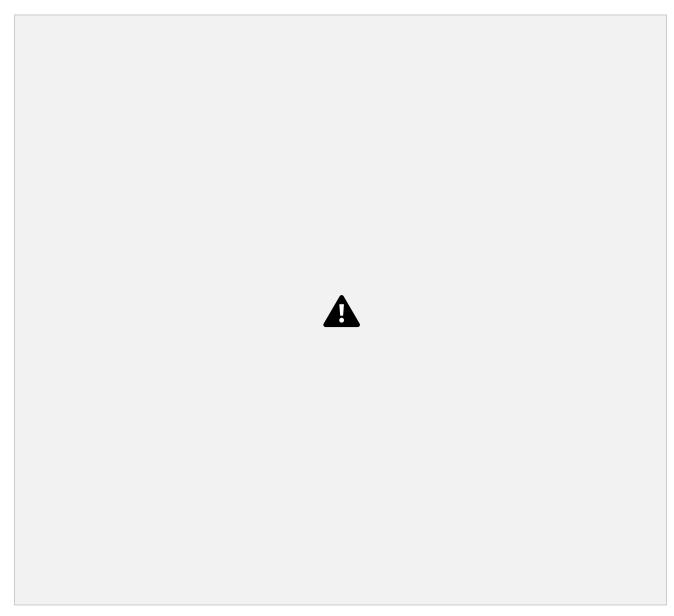




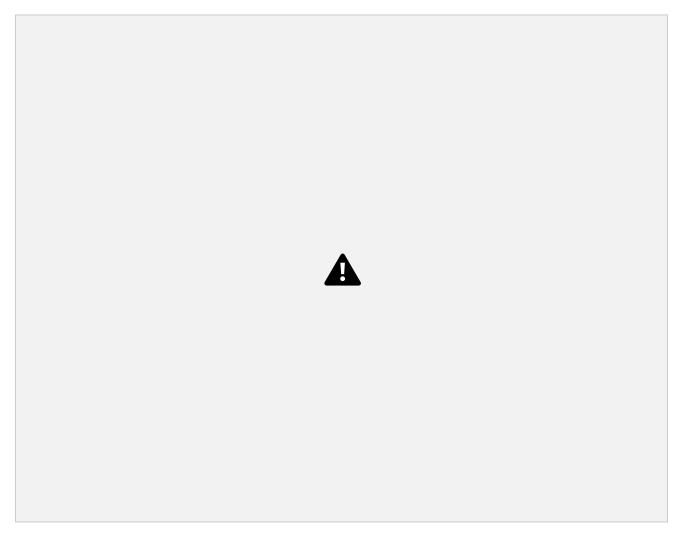


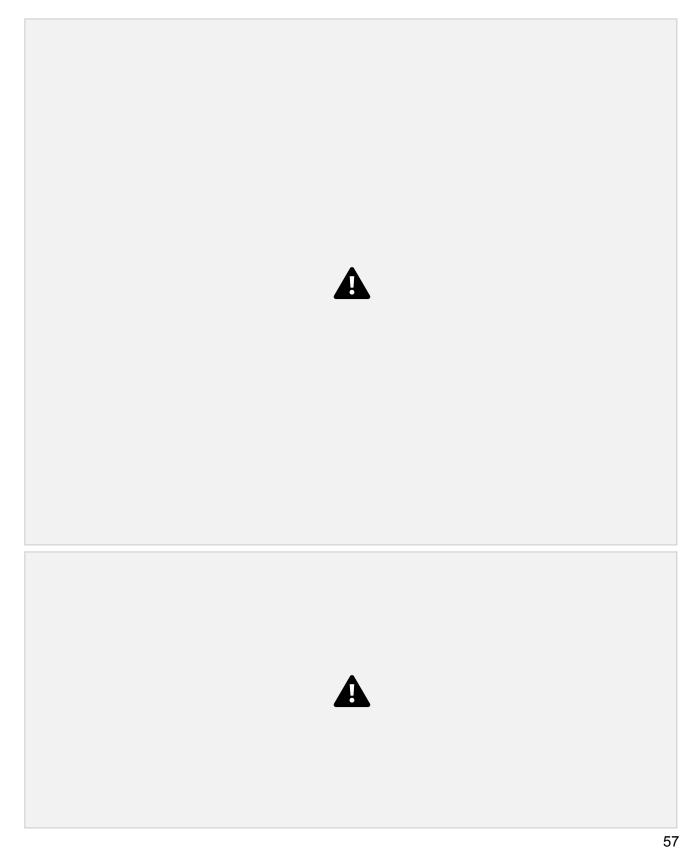


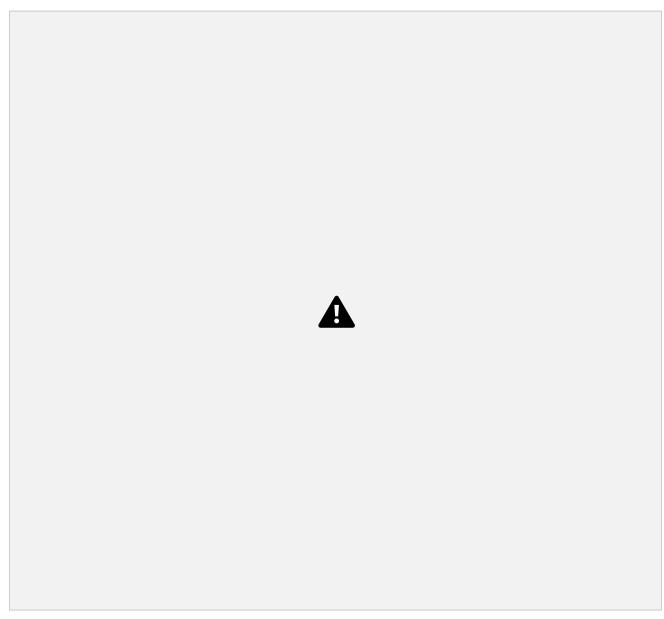


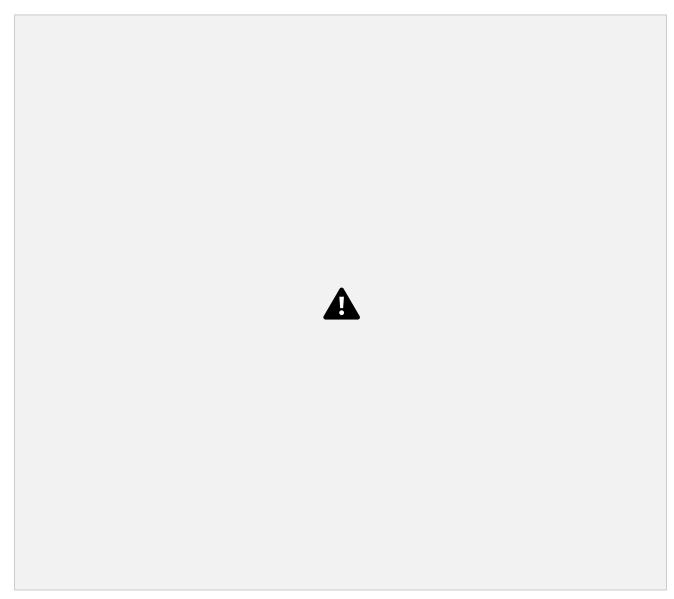










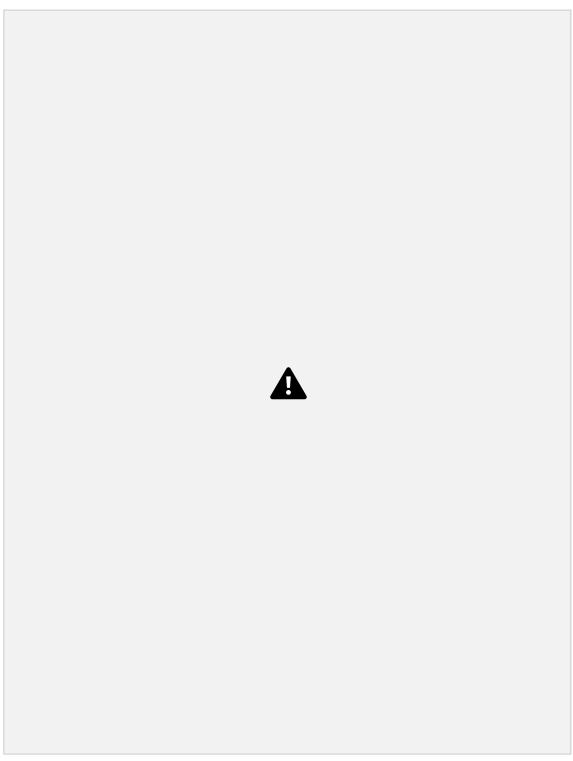


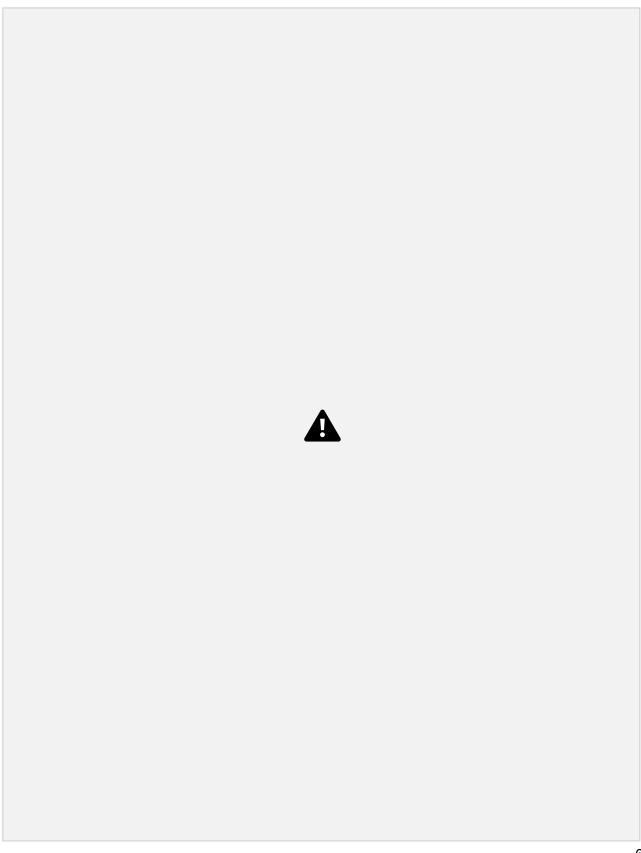


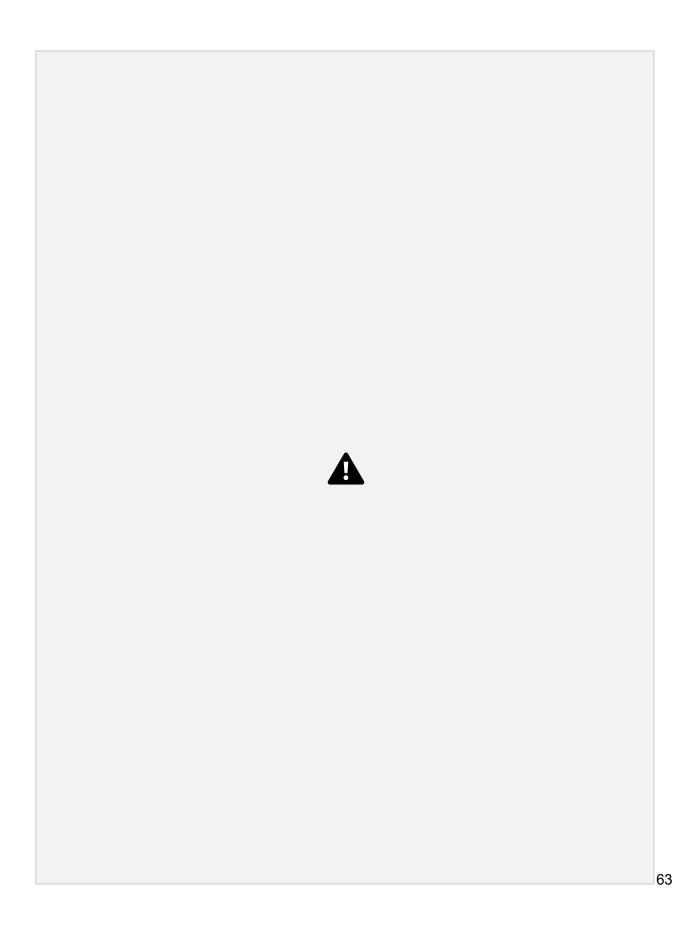
60

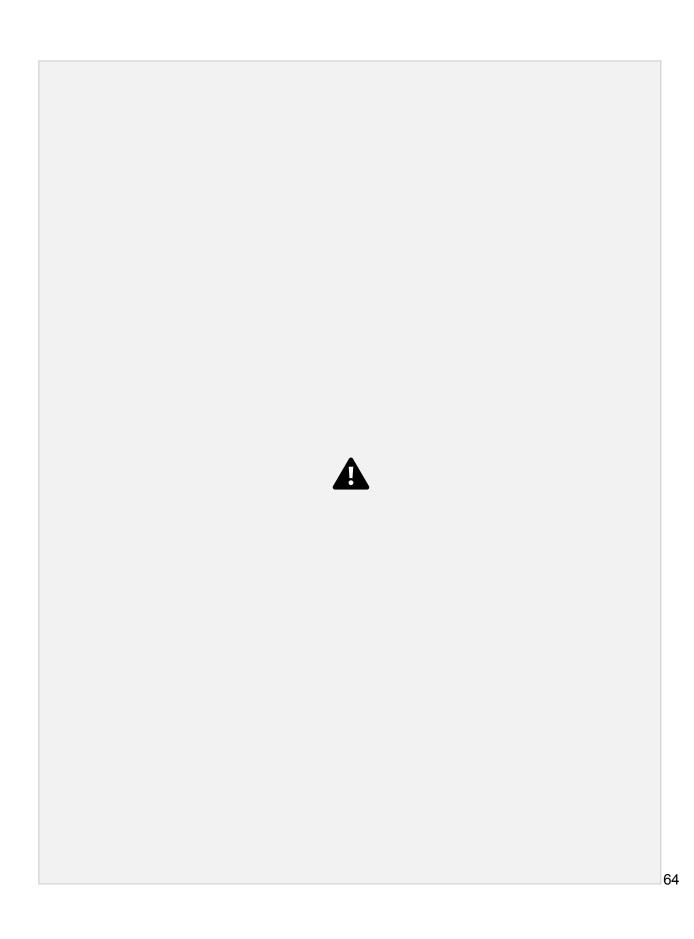
- Program 6(a) WAP to Implement Single Link List with following operations:-Sort the linked list,

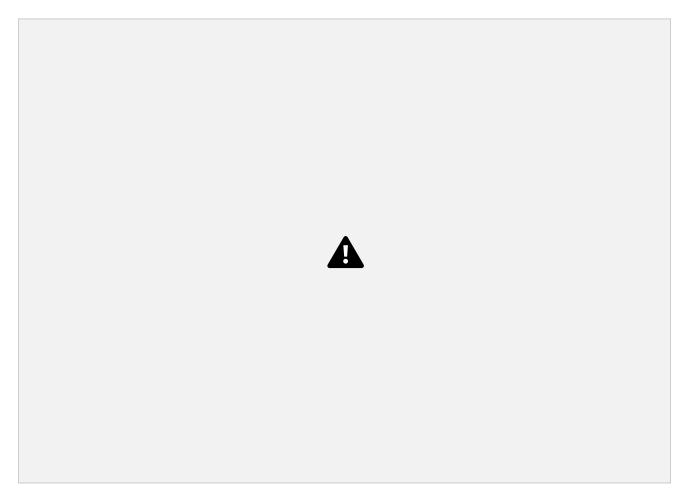
 - -Reverse the linked list,
 - -Concatenation of two linked lists.

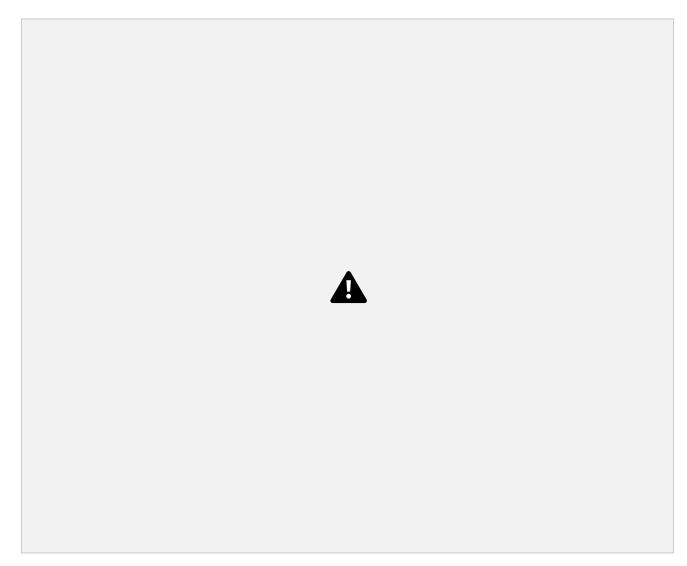


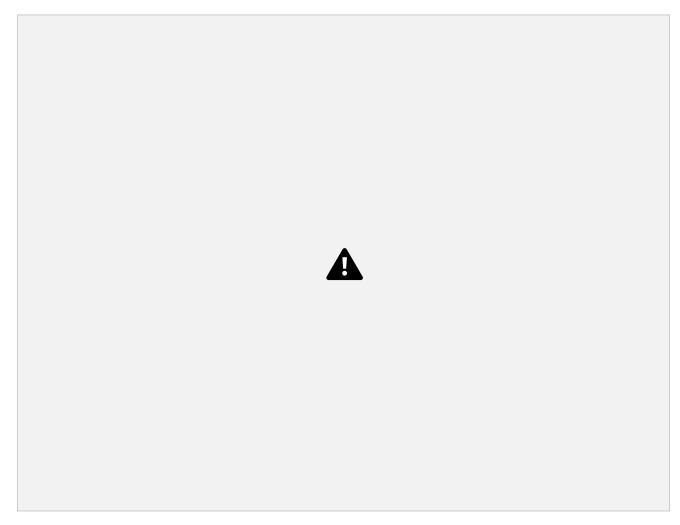


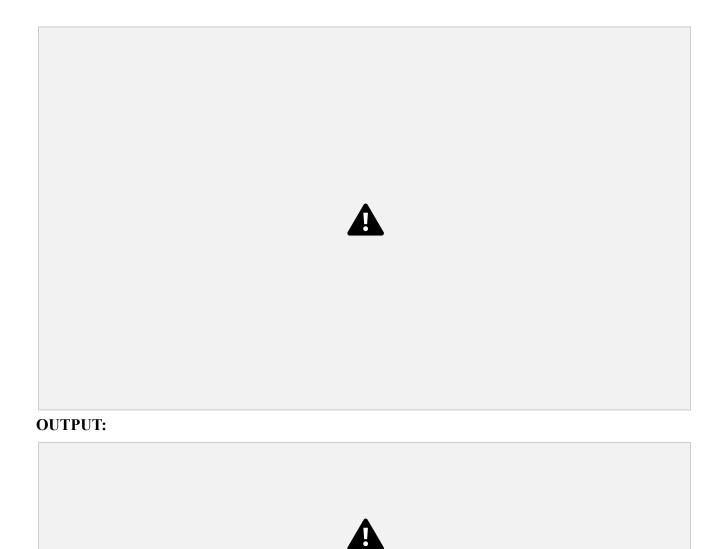






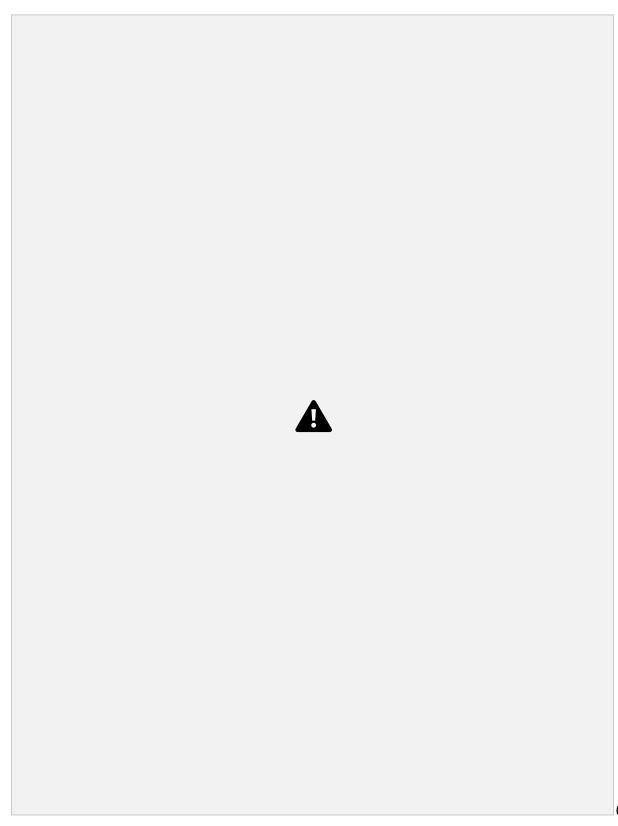


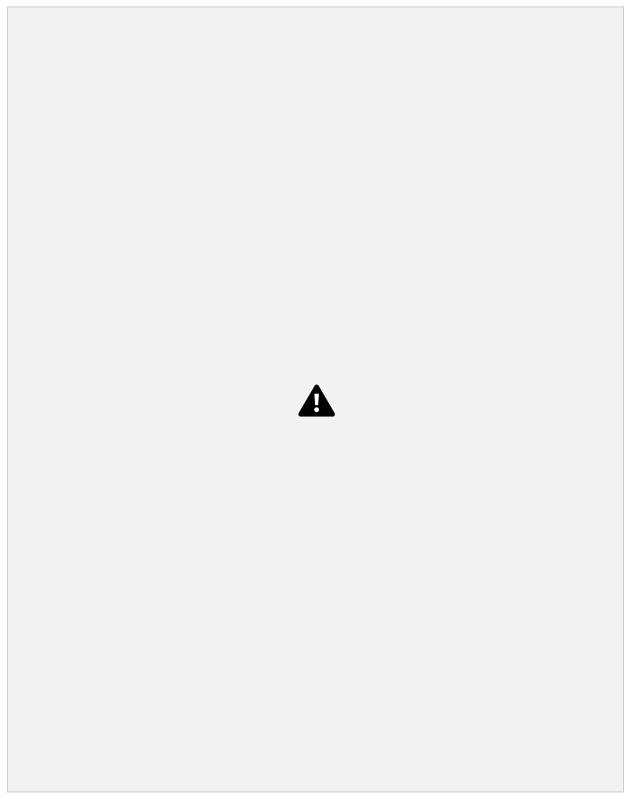


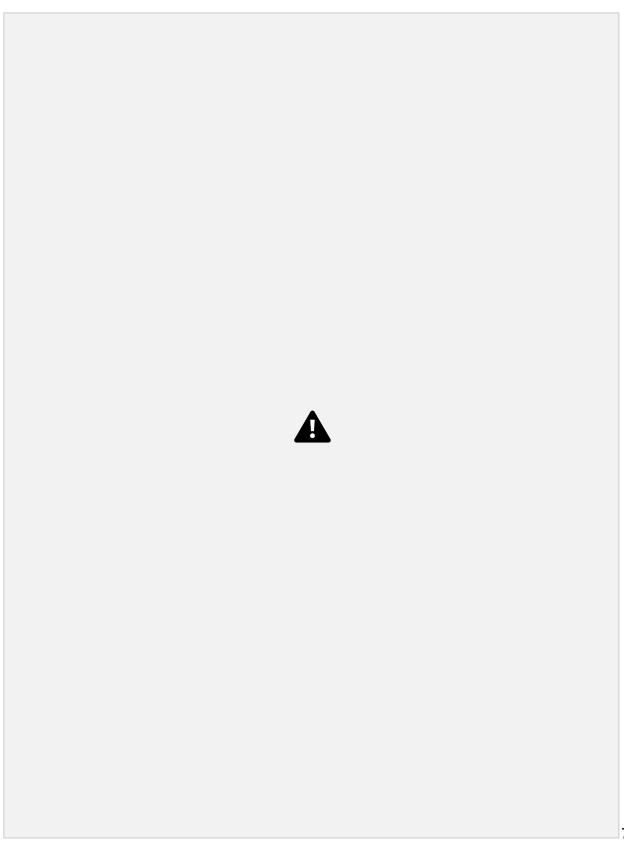


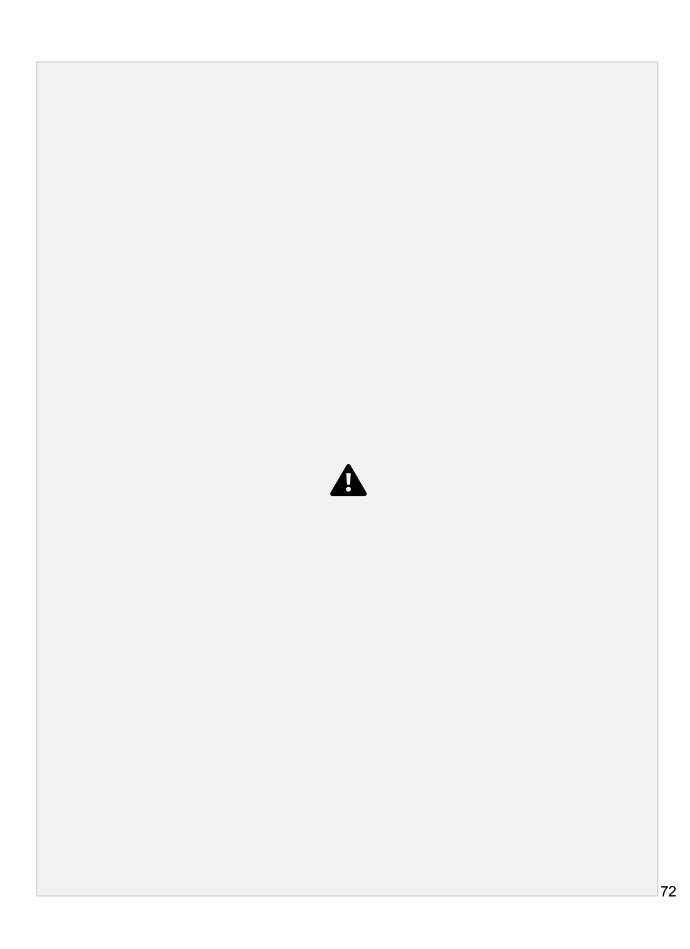
68

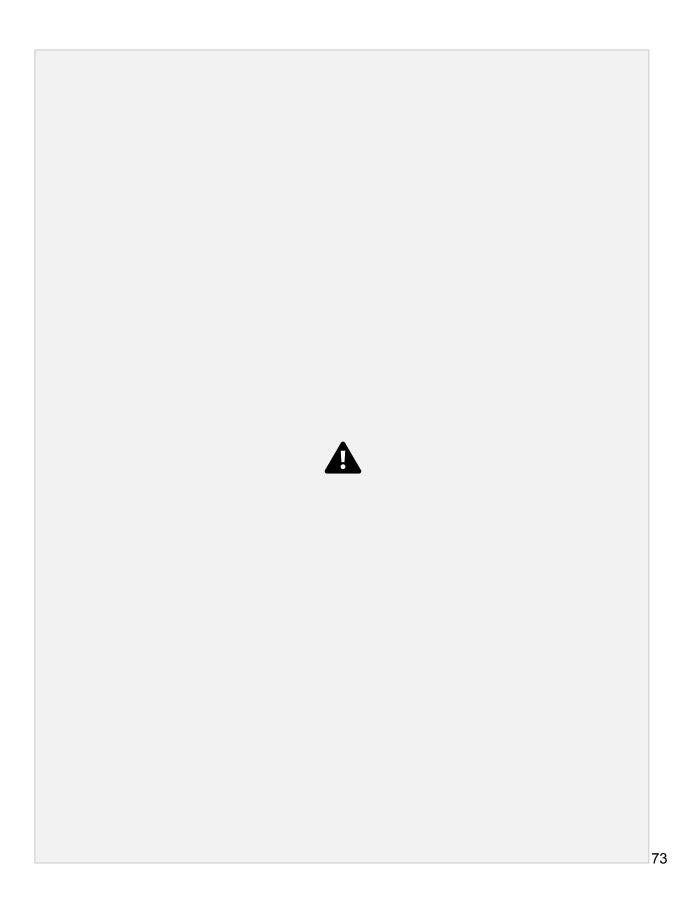
(b) WAP to Implement Single Link List to stimulate Stack and Queue operations.

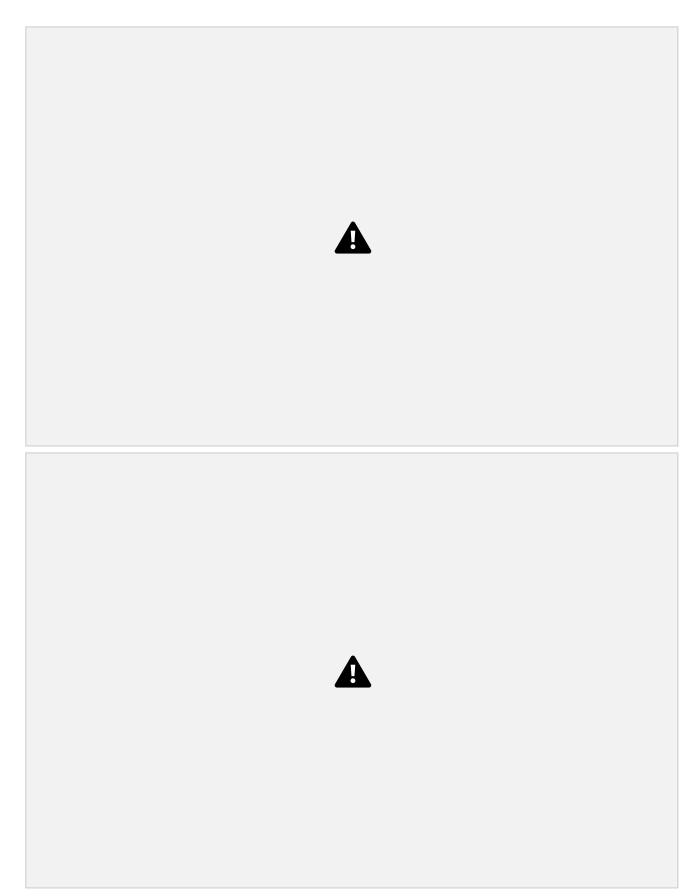


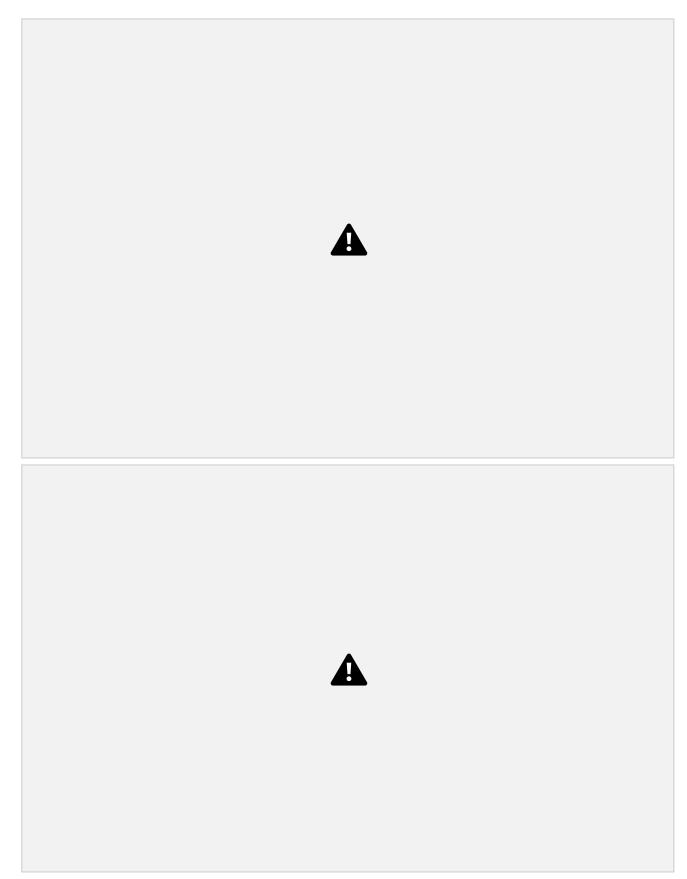










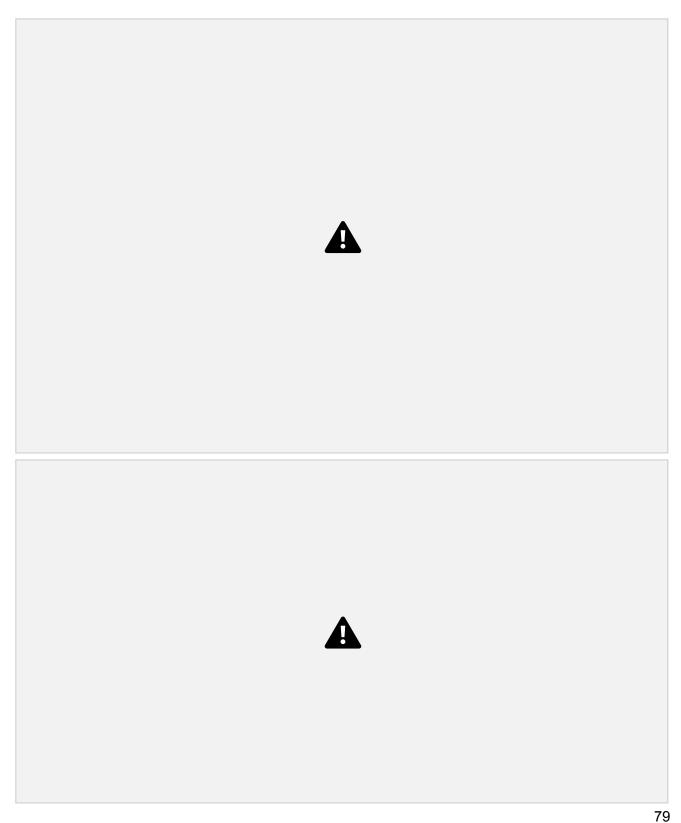












Program 7

WAP to implement doubly linked list with operations:

(a) Create a doubly linked list

(b) Insert a new node to the left of the node

