VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.



LAB REPORT on

Data Structures using C Lab

(23CS3PCDST)

Submitted by

Sharada Koundinya (1BM23CS310)

in partial fulfilment 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 **Sharada Koundinya**(**1BM23CS310**), who is bona fide student of **B.M.S. College of Engineering.** It is in partial fulfilment 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.

Namratha S Assistant Professor Department of CSE, BMSCE Dr. Kavitha Sooda Professor &HOD Department of CSE, BMSCE

Index

Sl. No.	Date	Experiment Title	Page No.
1	30-09-24	working of stack using an array	4-6
2	07-10-24	convert a given valid parenthesized infix arithmetic expression to postfix expression	7-9
3	14-10-24	A. working of a queue of integers using an array	10-12
	21-10-24	B. working of a circular queue of integers using an array	13-15
4	28-10-24	Implement Singly Linked List with following operations a) Create a linked list. b) Insertion of a node at first position, at any position and at end of list. Display the contents of the linked list.	16-20
5	11-10-24	Implement Singly Linked List with following operations a) Create a linked list. b) Deletion of first element, specified element and last element in the list. c) Display the contents of the linked list.	21-25
6	2-12-24	A. Implement Single Link List with following operations: Sort the linked list, Reverse the linked list, Concatenation of two linked lists.	25-29
		B. Implement Single Link List to simulate Stack & Queue Operations.	29-32
7	16-12-24	Implement doubly link list with primitive operations	33-37
8	23-12-24	Write a program a) To construct a binary Search tree. b) To traverse the tree using all the methods i.e., in-order, preorder and post order c) To display the elements in the tree.	37-39
9	23-12-24	A. traverse a graph using BFS method.B. check whether given graph is connected or not using DFS method.	40-42 43-44

Github Link:

https://github.com/sharadakoundinya/1BM23CS310_sharadakoundinya_dslab

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

```
#include<stdio.h>
#include<stdlib.h>
#define size 5
int top=-1;
int stack[size];
int item;
void push(){
  if(top==size-1){
     printf("Stack Overload\n");
  else{
    top+=1;
     stack[top]=item;
  }
}
int pop(){
  if(top==-1){
    printf("Stack Underflow\n");
  }
  else{
     return stack[top--];
  }
}
void display(){
  if(top==-1){
    printf("Stack is empty!");
  }
```

```
else{
     printf("Content of the stacks:");
     for(int i=0;i \le top;i++)
       printf("%d ",stack[i]);
     }printf("\n");
  }
void main(){
  int choice;
  while(1){
    printf("Enter your options:\n");
     printf("1.Push\n2.Pop\n3.Display\n4.Exit\n");
    printf("Enter your choice:");
     scanf("%d",&choice);
     switch(choice){
       case 1:printf("Enter the element to be pushed in:");scanf("%d",&item);push();break;
       case 2:if(top==-1){
          printf("stack is empty!\n");
       }else{
          printf("%d popped from stack\n", stack[top]);
       pop();
       break;
       case 3:display();
       break;
       case 4:exit(0);
  Enter your choice:1
  Enter the element to be pushed in:45
  Enter your options:
  1.Push
  2.Pop
  3.Display
  4.Exit
  Enter your choice:1
  Enter the element to be pushed in:67
  Enter your options:
  1.Push
  2.Pop
  3.Display
  4.Exit
  Enter your choice:3
  Content of the stacks:45 67
  Enter your options:
  1.Push
  2.Pop
  3.Display
  4.Exit
  Enter your choice:2
  67 popped from stack
  Enter your options:
  1.Push
  2.Pop
  3.Display
  4.Exit
  Enter your choice:
```

Stock	Operations
push (item)	SIZE
biny (green)	arely 312C the sale
	1) top = (SIZE -1)
	(- (+ - 1 - 1 - 1) IS
exit	and make
3	1 sub
else	sultry nearlys
3	
dep = Jop + 1	
stock [top] = it	nom
else 3 Step = step + 1 Stark [top] = st	top=-1
s pop (item)	Tingle
8	to anich dung esta
if stack is a	mply top = = 1
7.	0
exit 3	Why Breather Busy Police
	Pub
solve s	901
	[too]
item = stack	Tap
1-get = get	and the first first the same of
metis neutore	The same of the sa
3	shiply and I lot A
peor()	and of Structural
S	a Committee of the Comm
notum stack	[det]
gretuen stack	wall borned the
) Must in load 3	
9	

1	Date
	rays
-	if (top == MAXSIZE)
	grotum towe;
	stetion toue; ohe setion flave;
	bed is empty (i)- ii (tep == -1) Strion toue;
	tool is empty ()
	ij (top == -1)
	return true;
	ohe
	setion take;
	3
7 7	1 + get = and
reen	that a Lad year
7 1	
	OUTPUT)
	Output Gazate Gazate
	Enter your choice: 1
	Sites the element to be pushed in: 45
	Enter your options
	1. Push
	2. Pap
	3. Duday
	4. Exit foot sot = mot
	Enter your choice: 1
	Enter element to be pushed in: 67
	Entos your choice: 3
	Contents of the stack are 45 67
	D MAL WE 40 61
	Entre inia alainia
	Enter your choice: 2 67 pepper your stack
	6 popped from stack
	Other was tred
	N A A A A A A A A A A A A A A A A A A A
	A COLUMN THE PARTY OF THE PARTY

Write A Program to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide)

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#define MAX 5
char stack[MAX];
int top = -1;
void push(char c) {
  if (top < MAX - 1) {
     stack[++top] = c;
  }
char pop() {
  if (top >= 0) {
     return stack[top--];
  return '\0';
char peek() {
  if (top >= 0) {
     return stack[top];
  return '\0';
int precedence(char c) {
  switch (c) {
     case '+': return 1;
     case '-': return 1;
     case '*': return 2;
     case '/': return 2;
     case '^': return 3;
     default: return 0;
  }
}
```

```
int isOperator(char c) {
  return c == '+' \parallel c == '-' \parallel c == '*' \parallel c == '/' \parallel c == '^';
}
void infixToPostfix(const char *infix, char *postfix) {
  int i = 0, j = 0;
  while (infix[i]) {
     if (isalnum(infix[i])) {
        postfix[j++] = infix[i];
     } else if (infix[i] == '(') {
        push(infix[i]);
     } else if (infix[i] == ')') {
        while (top != -1 && peek() != '(') {
           postfix[j++] = pop();
        pop();
     } else if (isOperator(infix[i])) {
        while (top != -1 && precedence(peek()) >= precedence(infix[i])) {
           postfix[j++] = pop();
        push(infix[i]);
     i++;
  while (top !=-1) {
     postfix[j++] = pop();
  postfix[j] = '\0';
int main() {
  char infix[MAX], postfix[MAX];
  printf("Enter an infix expression: ");
  scanf("%s", infix);
  infixToPostfix(infix, postfix);
  printf("Postfix expression: %s\n", postfix);
  return 0;
}
```

Enter an infix expression: abcd^e-fgh*+^*+i-Postfix expression: abcde^fgh*-^*+i+-

```
the well to convert a good valid possitions in fix as with make expression to pastifix expression.

# linclude < statio. h>
# linclude < statio. h

#
```

Page	Date_/_/
1 (! special !! Latack) & waster !! (! hospital ! Ap. ")	Page AC OI AI
paint ("Hemory allexation pailed! In").	stack[++ stack Index] = c.
paint (Memory attended	2 anana ananah a anana 6
gietusin;	shile (starkIndex >= 0) &
3 XX ailtos abidocilát	- tack Istack Inter-
Jen (int i = 0; [< lon; i++) & student + chor c = 8[i];	? Stessed [Stessed Index++] = Black [Stack Indx-]
clear c = Stil.	result [result Index] = '10';
Office Sees)	Definite ("28") Itmen (
11 (c>='a' ff c<='z')11 (c>='A' ff c<='z') 11 (c>='0' ff c<='9')12	frank () S III , Siesall /;
11(C)=10' le C = 19') }	1900 (90mit).
2 - Little Communication of the Communication of th	free (Secult);
	mare the second
dre if (c = '()' & market	int main() &
the state of the s	choe 200[] = "a,b*(c^d-e)^(+,0*1)
stack[++ stackIndex]=C;	$\frac{(a+g^*h)}{(a+g^*h)}$
2	
ohe ij (c == ")) & mater	injux ToPartfix (exp.);
while (stack Take >= 0 Pl stack (stack Take Take)	and romagn caps,
while (stack Index >= 0 ft stack [stack Index]!	eretuen O;
grewitt [gewittindex ++] = stack [stack Index-]	3
3 Siesau (Apana Hoex + 19 Saint Island Hoex + 19	outot
stackIndex; I wouldnot	output
Suckinger ;	abcd^e-lgh*+/*+i-
olse S.	, N
	Narada 1 or
while (stackIndex >= 0 gl (psiec (c) <	1 4 1/0/2
poiec (stack[stackIndex]) (poiec (c) == poiec	oulou = Trangle Landing
(stack Istack Index) ff associativity (c) == (L')))	(9) DV " quant of habitati to to 11
+ (== (6))) + entr/ = thurs + prob	
· (108) sallown (* perts) = stock 10	
gresult [gresult Index ++] = [stack Index-];	Sugurah boy
3 Alexander Control	
	() strong 1) B

Program 3a

Write A Program to simulate the working of a queue of integers using an array. Provide the following operations: Insert, Delete, Display

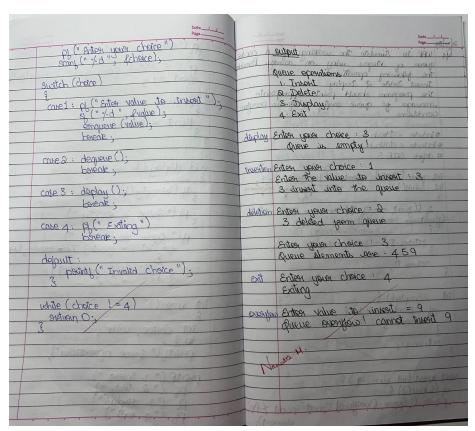
The program should print appropriate messages for queue empty and queue overflow conditions

```
#include <stdio.h>
#define max size 4
int queue [max_size];
int front =-1;
int rear=-1;
void insert(int value){
  if (rear==max_size -1){
     printf("Queue overflow! Cannot insert elements");
  else{
     if(front == -1)
       front =0;
     queue[++rear]=value;
    printf("Insert %d into queue",value);
  }
}
void delete(){
  if(front==-1 || front>rear){
     printf("Queue underflow!Cannot delete ");
  else{
     printf("Deleted %d from the queue",queue[front]);
    front++;
void display(){
  if (front==-1 || front>rear){
     printf("Queue is empty");
  else{
     printf("Queue Elements\n");
     for(int i=front;i<=rear;i++){</pre>
       printf("%d ",queue[i]);
```

```
printf("\n");
int main(){
  int choice, value;
  while(1){
     printf("\n1.Insert");
     printf("\n2.Delete");
     printf("\n3.Display");
     printf("\n4.Exit");
     printf("\nEnter your choice:");
     scanf("%d",&choice);
     switch (choice){
       case 1: printf("Enter a value to insert:");
             scanf("%d",&value);
             insert(value);
            break;
       case 2: delete();
       break;
       case 3: display();
       break;
       case 4: return 0;
       default: printf("Invalid choice! Please try again\n");
  }
```

```
Enter your choice:1
Enter a value to insert:4
Insert 4 into queue
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:1
Enter a value to insert:5
Insert 5 into queue
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:1
Enter a value to insert:9
Insert 9 into queue
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:1
Enter a value to insert:9
Insert 9 into queue
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:3
Queue Elements
4 5 9
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:4
```

B3 upp to stimulate the weeking of a queue of integent using an animay	of (" Queue is ampty");
a driene of intelleges month	
int queue MAX: votal total tallitus	go (" of deleted from queue," queue (gont)
not queue (MAX); which the state of the stat	al Rout > stead
setuen steed = max = 1 3 / may	
	2 (sont = 91809 = -1.
Int hempty()	3
entury gont == -1; 11 front > siens.	yeld diplay ()
	7
void ampione (not value)	ef (icempty ())
7	Pt (" Quave is empty")
of (while ()) of good astrolot you	
spoont (" Queue exempleso")	of (" queue alements: ")
() () () () () () () () () () () () () (
$\frac{3}{3}(\text{Konl} == -1) \qquad \text{tigtus}$ $\frac{1}{3}(\text{Konl} == -1) \qquad \text{tigtus}$ $\frac{1}{3}(\text{Konl} == 0) \qquad \text{tigtus}$	Pf (" old ," queue [i])
180nt = 0;	
97000 + 1	int man ()
queue [succer] = value; (If (" & d invested to queue ", value);	int chaice, value;
pr (" % of invested to greve ", value);	8 est
	parinty (" Queue operations)
reid dequeue	pointf (" 1. invest ")
	(" tresen" .1 ") Itriucq (" 3teleb .6 ") 19
siz (is empty ())	PL ("3. display") PL ("4. exit")



Program 3b

Write A Program to simulate the working of a circular 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

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 5
int queue[MAX];
int front = -1;
int rear = -1;
int isFull() {
  return (front == (rear + 1) % MAX);
}
int isEmpty() {
  return (front == -1);
void insert(int value) {
  if (isFull()) {
    printf("Queue Overflow: Unable to insert %d\n", value);
     return;
  if (isEmpty()) {
     front = 0; // Set front to 0 if the queue is empty
  rear = (rear + 1) \% MAX;
  queue[rear] = value;
  printf("Inserted %d into the queue\n", value);
void delete() {
  if (isEmpty()) {
    printf("Queue Underflow: Unable to delete from the queue\n");
    return;
  }
  int deletedValue = queue[front];
  if (front == rear) {
     front = -1; // Queue becomes empty
    rear = -1;
```

```
} else {
     front = (front + 1) \% MAX;
  printf("Deleted %d from the queue\n", deletedValue);
void display() {
  if (isEmpty()) {
     printf("Queue is empty\n");
     return;
  printf("Queue elements: ");
  int i = front;
  while (1) {
     printf("%d ", queue[i]);
     if (i == rear) break;
     i = (i + 1) \% MAX;
  printf("\n");
}
int main() {
  int choice, value;
  while (1) {
     printf("\nCircular Queue Operations:\n");
     printf("1. Insert\n");
     printf("2. Delete\n");
     printf("3. Display\n");
     printf("4. Exit\n");
     printf("Enter your choice: ");
     scanf("%d", &choice);
     switch (choice) {
       case 1:
          printf("Enter value to insert: ");
          scanf("%d", &value);
          insert(value);
          break;
       case 2:
          delete();
          break;
       case 3:
          display();
          break;
       case 4:
          exit(0);
```

```
default:
    printf("Invalid choice. Please try again.\n");
}

return 0;
}

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 85
Inserted 85 into the queue

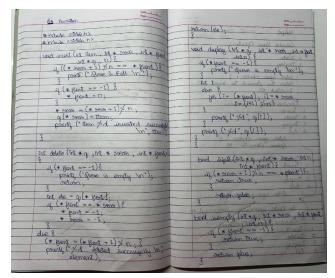
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 56
Inserted 56 into the queue

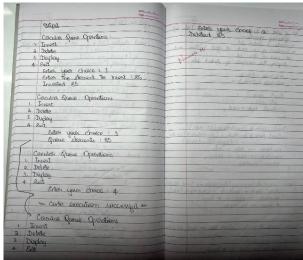
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 85 56

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 85 56

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 85 56
```

Enter your choice: 2 Deleted 85 from the queue





Write A Program to Implement Singly Linked List with following operations

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

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
  int data;
  struct Node* next;
};
void createList(struct Node** head);
void insertAtBeginning(struct Node** head, int data);
void insertAtPosition(struct Node** head, int data, int position);
void insertAtEnd(struct Node** head, int data);
void displayList(struct Node* head);
int main() {
  struct Node* head = NULL;
  int choice, data, position;
  while (1) {
     printf("\nMenu:\n");
     printf("1. Create a linked list\n");
     printf("2. Insert at the beginning\n");
     printf("3. Insert at a specific position\n");
     printf("4. Insert at the end\n");
     printf("5. Display the list\n");
     printf("6. Exit\n");
     printf("Enter your choice: ");
     scanf("%d", &choice);
     switch (choice) {
       case 1:
          createList(&head);
          break;
       case 2:
          printf("Enter data to insert at the beginning: ");
          scanf("%d", &data);
```

```
insertAtBeginning(&head, data);
          break;
       case 3:
          printf("Enter data to insert: ");
          scanf("%d", &data);
          printf("Enter position to insert (starting from 1): ");
          scanf("%d", &position);
          insertAtPosition(&head, data, position);
          break;
       case 4:
          printf("Enter data to insert at the end: ");
          scanf("%d", &data);
          insertAtEnd(&head, data);
          break;
       case 5:
          displayList(head);
          break;
       case 6:
          printf("Exiting the program.\n");
          exit(0);
       default:
          printf("Invalid choice. Please try again.\n");
  return 0;
void createList(struct Node** head) {
  int data, choice;
  do {
     printf("Enter data to insert: ");
     scanf("%d", &data);
    insertAtEnd(head, data);
     printf("Do you want to add another node? (1 for Yes, 0 for No): ");
     scanf("%d", &choice);
  \} while (choice != 0);
}
void insertAtBeginning(struct Node** head, int data) {
  struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  if (newNode == NULL) {
    printf("Memory allocation failed.\n");
    return;
  newNode->data = data;
  newNode->next = *head;
```

```
*head = newNode;
  printf("Node inserted at the beginning.\n");
void insertAtPosition(struct Node** head, int data, int position) {
  if (position < 1) {
    printf("Invalid position.\n");
    return;
  }
  struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  if (newNode == NULL) {
    printf("Memory allocation failed.\n");
    return;
  newNode->data = data;
  if (position == 1) {
    newNode->next = *head;
     *head = newNode:
    printf("Node inserted at position %d.\n", position);
    return;
  }
  struct Node* temp = *head;
  for (int i = 1; i < position - 1 && temp != NULL; <math>i++) {
    temp = temp->next;
  }
  if (temp == NULL) {
    printf("Position out of bounds.\n");
    free(newNode);
    return;
  }
  newNode->next = temp->next;
  temp->next = newNode;
  printf("Node inserted at position %d.\n", position);
void insertAtEnd(struct Node** head, int data) {
  struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  if (newNode == NULL) {
    printf("Memory allocation failed.\n");
    return;
  newNode->data = data;
```

```
newNode->next = NULL;
  if (*head == NULL) {
    *head = newNode;
  } else {
    struct Node* temp = *head;
    while (temp->next != NULL) {
       temp = temp->next;
    temp->next = newNode;
  printf("Node inserted at the end.\n");
void displayList(struct Node* head) {
  if (head == NULL) {
    printf("The list is empty.\n");
    return;
  printf("Linked list contents: ");
  struct Node* temp = head;
  while (temp != NULL) {
    printf("%d -> ", temp->data);
    temp = temp->next;
  printf("NULL\n");
```

```
Menu:

1. Create a linked list
2. Insert at the beginning
3. Insert at a specific position
4. Insert at the end
5. Display the list
6. Exit
Enter your choice: 2
Enter data to insert at the beginning: 12
Node inserted at the beginning.

Menu:
1. Create a linked list
2. Insert at the beginning
3. Insert at a specific position
4. Insert at the end
5. Display the list
6. Exit
Enter your choice: 2
Enter data to insert at the beginning: 23
Node inserted at the beginning.

Menu:
1. Create a linked list
2. Insert at the beginning.

Menu:
1. Create a linked list
2. Insert at the beginning
3. Insert at a specific position
4. Insert at the end
5. Display the list
6. Exit
Enter your choice: 3
Enter data to insert: 2
Enter position to insert (starting from 1): 2
Node inserted at position 2.
```

```
Menu:
1. Create a linked list
2. Insert at the beginning
3. Insert at a specific position
4. Insert at the end
5. Display the list
6. Exit
Enter your choice: 5
Linked list contents: 23 -> 2 -> 12 -> NULL
Menu:
1. Create a linked list
2. Insert at the beginning
3. Insert at a specific position
4. Insert at the end
5. Display the list
6. Exit
Enter your choice: 6
Exiting the program.
```

Date	Date 1 1
25 of MAR to Tomperment Singly Linked Lut us Glowing operations	*herd = newNote.
	The contract of the second
investion of a node at food position and	to list i = 1 * temp = * hand;
at and of the list authorized lists authorized lists	ten in i cost of semple MULL in it
# include < stdio.h>	y (Jenny == quret) ju
#include < statio.h >	3 pount (" Position out of month (");
stouct Node &	(3 dee El) method to hour
int data;	newhode → next = temp → rext;
Sourt Note * next;	Jamp → next = newNode;
3;	2
struct Node * coverteNode (int data) {	void invest at end (struct Node ** head,
struct Node * pewNode = (structNode *) maller	Sot data) &
(shall trunk) lead)	stantage = shall can * shall trunte
newNede → data = data;	(data).
newhode -> next = NULL;	if (* head == NULL) }
sietusin neioNode;	* head = newNede;
3	3 Subtuans
void invest at beginning (Journale ** head, Pit 1871)	struct Node * temp = * head
stouct Node * neur Node = Opposter Node (data);	while (temp - rest = Now)?
nowhole → next = * head;	trans commote = quart
* head = newNode;	to cot - was a lodge to
5	3 temp - next = new Made;
1 3 2 3 2 4 5 13 1 3 4 5	Thorn & shall winter be all the state of the
void investal partien (stouch who we had intented	11 (1-20 4 == VIII 1) \$
	wenter of the second (" country setted by").
sp (pob == 1) &	eague de par print ("Empire atil In");
newNode -> next = * head;	S TO WASHINGTON
	3

Appe	Date/_
Sourt Nade * temp = head; while (temp! = NNU!) & pourty (' d - temp - data) temp = temp - rext;	Enter your choice 3 Jinhad July 12 - 23 - HULL
while (temp! = NULL) &	Jinhad Sit 112 - 23 - NUM
point (of d -> temp -> data)	
temp = temp = rest;	enter your choice: 1
(" mul \m");	winds . As
3 point ("NIT IN");	Enter your choice: 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2	2ntex: 568
John Many & head = MULL	and the same of th
	Enter your choice 3
invest at beginning (fread, 5); display (head);	enton:
display (head);	Jinted Jist: 98 → 12 → 33 → 568
event + and (Phond 15 3).	(1: Vallant Int. 14 (1 == [1])
display (head); invest at pess (shead, 15,3); display (head);	
all the trade to the first train that	o e ree
must at end (thend, 30);	
diplay (head); I am trade	and the second second
/ // //	
3 return 0;	
3 shallman = trait	M. 13.
morten	70,707
output.	19-7
and & - youth & still town	
1. Truest at the sheginning	
a. Treest at the end	
3. Diplay the Jul	
4. Exit March transformat	
Enter your choice: 1	
Enter value to invest at the beginning to	THE RESERVE TO A STATE OF THE PARTY OF THE P
Enter your choice: 2 mg	and expensely the second
Entor your choice: 2 9	
Enter the value to insort at the end 28	

Write A Program to Implement Singly Linked List with following operations

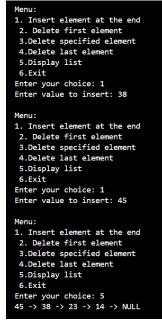
- a) Create a linked list.
- b) Deletion of first element, specified element and last element in the list.
- c) Display the contents of the linked list.

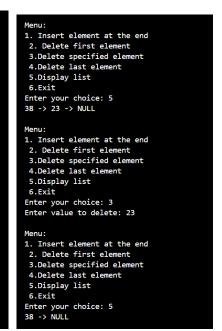
```
#include <stdio.h>
#include <stdlib.h>
struct Node {
  int data;
  struct Node* next;
};
struct Node* createNode(int data) {
  struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  newNode->data = data:
  newNode->next = NULL;
  return newNode;
};
void insertatfirst(struct Node** head, int data){
  struct Node* newnode =createNode(data);
  newnode->next = *head;
  *head = newnode;
}
void deleteFirst(struct Node** head) {
  if (*head == NULL) {
    printf("The list is empty.\n");
    return;
  struct Node* temp = *head;
  *head = (*head)->next;
  free(temp);
}
void deleteElement(struct Node** head, int key) {
  if (*head == NULL) {
    printf("The list is empty.\n");
    return;
  struct Node *temp = *head, *prev = NULL;
  if (temp != NULL && temp->data == key) {
```

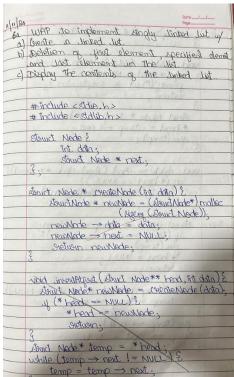
```
*head = temp->next;
    free(temp);
    return;
  }
  while (temp != NULL && temp->data != key) {
    prev = temp;
    temp = temp->next;
  if (temp == NULL) {
    printf("Element %d not found.\n", key);
    return;
  prev->next = temp->next;
  free(temp);
void deleteLast(struct Node** head) {
  if (*head == NULL) {
    printf("The list is empty.\n");
    return;
  struct Node *temp = *head, *prev = NULL;
  if (temp->next == NULL) {
    *head = NULL;
    free(temp);
    return;
  while (temp->next != NULL) {
    prev = temp;
    temp = temp->next;
  prev->next = NULL;
  free(temp);
void displayList(struct Node* head) {
  if (head == NULL) {
    printf("The list is empty.\n");
    return;
  }
  struct Node* temp = head;
  while (temp != NULL) {
    printf("%d -> ", temp->data);
    temp = temp->next;
  printf("NULL\n");
```

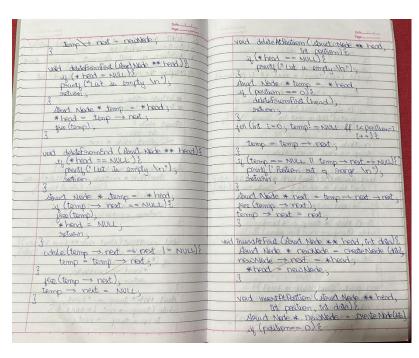
```
int main() {
  struct Node* head = NULL;
  int choice, value;
  while (1) {
    printf("\nMenu:\n");
     printf("1. Insert element at the end\n 2. Delete first element\n 3.Delete specified element\n
4.Delete last element\n 5.Display list\n 6.Exit\n");
     printf("Enter your choice: ");
    scanf("%d", &choice);
     switch (choice) {
       case 1:
          printf("Enter value to insert: ");
          scanf("%d", &value);
          insertatfirst(&head, value);
          break;
       case 2:
          deleteFirst(&head);
          break;
       case 3:
          printf("Enter value to delete: ");
          scanf("%d", &value);
          deleteElement(&head, value);
          break;
       case 4:
          deleteLast(&head);
          break;
       case 5:
          displayList(head);
          break;
       case 6:
          exit(0);
       default:
          printf("Invalid choice.\n");
  }
  return 0;
```

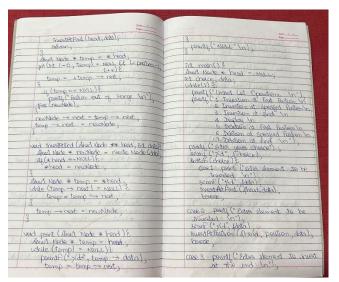
Menu: 1. Insert element at the end 2. Delete first element 3.Delete specified element 4.Delete last element 5.Display list 6.Exit Enter your choice: 2 Menu: 1. Insert element at the end 2. Delete first element 3.Delete specified element 4.Delete last element 5.Display list 6.Exit Enter your choice: 5 38 -> 23 -> 14 -> NULL Menu: 1. Insert element at the end 2. Delete first element 3.Delete specified element 4.Delete last element 5.Display list 6.Exit Enter your choice: 4

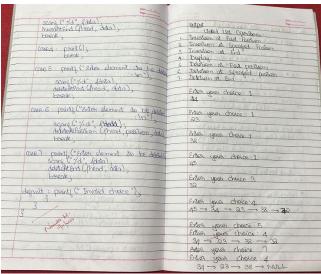












Program 6a

Write A Program to Implement Single Link List with following operations: Sort the linked list,
Reverse the linked list,
Concatenation of two linked lists.

```
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
  int data;
  struct Node* next;
} Node;
Node* createNode(int data) {
  Node* newNode = (Node*)malloc(sizeof(Node));
  if (!newNode) {
    printf("Memory error\n");
    return NULL;
  newNode->data = data;
  newNode->next = NULL;
  return newNode;
void insertNode(Node** head, int data) {
  Node* newNode = createNode(data);
  if (*head == NULL) {
```

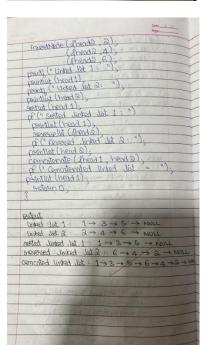
```
*head = newNode;
    return;
  Node* lastNode = *head;
  while (lastNode->next) {
    lastNode = lastNode->next;
  lastNode->next = newNode;
}
void printlist(Node* head) {
  Node* current = head;
  while (current) {
    printf("%d -> ", current->data);
    current = current->next;
  printf("NULL\n");
void sortlist(Node* head) {
  if (head == NULL) {
    return;
  Node* current;
  Node* nextNode;
  int temp;
  for (current = head; current != NULL; current = current->next) {
    for (nextNode = current->next; nextNode != NULL; nextNode = nextNode->next) {
       if (current->data > nextNode->data) {
         temp = current->data;
         current->data = nextNode->data;
         nextNode->data = temp;
void reverselist(Node** head) {
  Node* prev = NULL;
  Node* current = *head;
  Node* next = NULL;
  while (current) {
    next = current->next;
    current->next = prev;
    prev = current;
    current = next;
  }
```

```
*head = prev;
void concatenate(Node** head1, Node* head2) {
  if (*head1 == NULL) {
     *head1 = head2;
    return;
  Node* lastNode = *head1;
  while (lastNode->next) {
    lastNode = lastNode->next;
  lastNode->next = head2;
int main() {
  Node* head1 = NULL;
  Node* head2 = NULL;
  insertNode(&head1, 1);
  insertNode(&head1, 3);
  insertNode(&head1, 5);
  insertNode(&head2, 2);
  insertNode(&head2, 4);
  insertNode(&head2, 6);
  printf("Linked list 1: ");
  printlist(head1);
  printf("Linked list 2: ");
  printlist(head2);
  sortlist(head1);
  printf("Sorted Linked list 1: ");
  printlist(head1);
  reverselist(&head2);
  printf("Reversed Linked list 2: ");
  printlist(head2);
  concatenate(&head1, head2);
  printf("Concatenated Linked list: ");
  printlist(head1);
  return 0;
```

```
Linked list 1: 1 -> 3 -> 5 -> NULL
Linked list 2: 2 -> 4 -> 6 -> NULL
Sorted Linked list 1: 1 -> 3 -> 5 -> NULL
Reversed Linked list 2: 6 -> 4 -> 2 -> NULL
Concatenated Linked list: 1 -> 3 -> 5 -> 6 -> 4 -> 2 -> NULL
```

13	to lami at 940 harman
5	lit of the fell epolitions singly links
	sent the linked hit
	Stevelle The Linker lit
_	concalmating 2 linked lit
_	# Include < stip. h >
1	# include < stalib h>
	Struct Nede &
	int data.
	struct node * next.
-	3
	(b) Node (s) siller ha
-	Nade * createNade (int data) §
	NEWS * CHEMENISAS (int data) ?
	Nade * new Made = (Made *) mollec (size (Mode));
ī	IL (! new Nede) ?
Ī	count! ("Nomes made (n")-
j	and all all 1 :
	at (! new Alade)?. point (! Namery wood 'n"); potun NVLL;
	romNede -> data = data
	- newNade -> next = NULL
1	enetuen vieuNede.
	techt & dellar
Ť	yold investigate (Node ** head, int dat
I	- day Say = daystan
Ī	Nade * newhade = createNade (date
	* (* head == NULL) }
ĺ	* head = new Node;
İ	eretuen;
۲	2

Date	Date
while (IntNode > next)?	Elboard ** about Italaurayare bilay
while (lastNede = tastNede -> riest;	
	NEGE * CUNNIGHT - 41-1
latNode → next = newNode;	Node * noxt = NULL;
	Could Condidate 13
3 (host # bond) &	next = cusurent -> mad
void pointlist (Node * head) {	cusionent - next = poleis.
pewrd (", d -> ", head -> data); head = head -> next;	DAIEN = CHARLENT; PAIENS;
bood = bond -> next.	apprent = next.
	3 - 1/2 Breach Saltonice
point (" MULL 'n");	* head = paice;
Paint Note at a	3 1 1000 - 10000;
void southlist (Nexte # head) §	1 - 3 hand tult non
Note * current = head:	void compatenate (Node ** head?
at the sales	
if (part = MOLT) &	1) (hend 1 == 11111)S
entuern :	if (head1 == Null ? head 2)? * head1 = head2;
3	sietuoin.
Strile (consent) &	2
Nade * netNode = current > net.	Nede * latNede = * head 1
usule (nextNede) &	ustile (lost Node -> next) ?
While Cherthere's	JayNede = JoshAde → next
3 (consent → data → next Norte → data) {	Sandanas = National - Loxi
temp = cuspiont -> data;	
cunsient → data = nextibile-still	3 last bide \rightarrow next = heada;
rextNode -> data = temp;	C+ A-der Server tolk hours kalmines
how has a fair fall from her	
revtNede = nevtNede → next:	int main()
reatNade = rextrade → next;	Nede * head 1 = NULL;
cusiont = cusiquent → rext;	Nede * head a = NULL;
A Marie & Marie (Co.)	invest Node (shead) 1);
2	(Shead) 3)
J grandelle	(Shead1'5).
	(a) redit, si
- free 1 = shallted & shall	



Program 6b

Write A Program to Implement Single Link List to simulate Stack & Queue Operations.

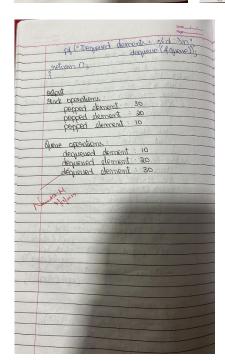
```
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
  int data;
  struct Node* next;
} Node;
Node* createNode(int data) {
  Node* newNode = (Node*)malloc(sizeof(Node));
  if (!newNode) {
    printf("Memory allocation error\n");
    return NULL;
  }
  newNode->data = data;
  newNode->next = NULL;
  return newNode;
void push(Node** top, int data) {
  Node* newNode = createNode(data);
  if (!newNode) return;
  newNode->next = *top;
  *top = newNode;
  printf("%d pushed to stack\n", data);
int pop(Node** top) {
  if (*top == NULL) {
    printf("Stack Underflow\n");
    return -1;
  Node* temp = *top;
  int poppedData = temp->data;
  *top = temp->next;
  free(temp);
  printf("%d popped from stack\n", poppedData);
  return poppedData;
void displayStack(Node* top) {
```

```
if (top == NULL) {
    printf("Stack is empty\n");
    return;
  }
  printf("Stack: ");
  Node* temp = top;
  while (temp) {
    printf("%d -> ", temp->data);
    temp = temp->next;
  printf("NULL\n");
void enqueue(Node** front, Node** rear, int data) {
  Node* newNode = createNode(data);
  if (!newNode) return;
  if (*rear == NULL) {
    *front = *rear = newNode;
  } else {
    (*rear)->next = newNode;
    *rear = newNode;
  printf("%d enqueued to queue\n", data);
int dequeue(Node** front, Node** rear) {
  if (*front == NULL) {
    printf("Queue Underflow\n");
    return -1;
  Node* temp = *front;
  int dequeuedData = temp->data;
  *front = temp->next;
  if (*front == NULL) {
    *rear = NULL;
  free(temp);
  printf("%d dequeued from queue\n", dequeuedData);
  return dequeuedData;
}
void displayQueue(Node* front) {
  if (front == NULL) {
    printf("Queue is empty\n");
    return;
  printf("Queue: ");
```

```
Node* temp = front;
  while (temp) {
    printf("%d -> ", temp->data);
    temp = temp -> next;
  printf("NULL\n");
int main() {
  Node* stackTop = NULL;
  printf("\n--- Stack Operations ---\n");
  push(&stackTop, 10);
  push(&stackTop, 20);
  push(&stackTop, 30);
  displayStack(stackTop);
  pop(&stackTop);
  displayStack(stackTop);
  Node* queueFront = NULL;
  Node* queueRear = NULL;
  printf("\n--- Queue Operations ---\n");
  enqueue(&queueFront, &queueRear, 1);
  enqueue(&queueFront, &queueRear, 2);
  enqueue(&queueFront, &queueRear, 3);
  displayQueue(queueFront);
  dequeue(&queueFront, &queueRear);
  displayQueue(queueFront);
  return 0;
 --- Stack Operations ---
 10 pushed to stack
 20 pushed to stack
 30 pushed to stack
 Stack: 30 -> 20 -> 10 -> NULL
 30 popped from stack
 Stack: 20 -> 10 -> NULL
 --- Queue Operations ---
 1 enqueued to queue
 2 enqueued to queue
 3 enqueued to queue
 Queue: 1 -> 2 -> 3 -> NULL
 1 dequeued from queue
 Queue: 2 -> 3 -> NULL
```

1	A solomi A ARCI
0	at till bestrik tilgnik der stock attimutet
/	at the bestral uppose transform at ARON sometiments
/	#include cotten by
/	Shurt Nede ?
	int data;
-	etauntalade * neut;
_	\$
	N/6do
	Nede, stanct stack &
	Nede * Jep;
	\$ 000
	stack:
	struct grove ? Nede * Sent;
	Node * sont.
	Node * Siegn.
	3 (Chart) ass
	Ougle.
	Node * createNade (int. data) &
	Node * newNode = (Node*) mollec
	Made * nounted = (Nede*) mallo (Mgr d, (Nede)) (Mgr d, (Nede))
	a (Incus Norte) ?
	the word (" Norman sone /");
	tange Will.
	geturn NULL;
4	
-	new Node -> etta next = NIVIL.
	new Node > children
	oreturn neurllode;
	3 man man a ctack) &
200	yord institlack (stack * stack) ?
	stock -> top = NULL;
	14 of = 1000 complete
-	

The termination of Cort * Goot * Good		
solven stack stack to data 2 solven from the stack to data 2 solven from the stack to the stac	Date	tota
solven stack stack to data 2 solven from the stack to data 2 solven from the stack to the stac	to the Charle & Stock &	ent dequare (Queue * augusts
west push (stort * stork int data) ? west push (stort * stork int data) ? west resultate = chanterbase (data). Ind data = queue * per data; freshede = next = stork = tep. gare = stork =	int 18 emphysica that == NULL:	2 ((group) groupy and F.
And reported the resultant training the state of the stat	ground said	Pl " Anone a arrety ").
And reported the resultant training the state of the stat	and out (stock * stock int data) &	set 1 section - 1;
Sant step = markete Sant math (Sant) & sant step Sant step = date Sant step = markete Sant step = date Sant step	Note * newNode = coventerNode (data).	in add = queue -> pent -> data.
Sant step = markete Sant math (Sant) & sant step Sant step = date Sant step = markete Sant step = date Sant step	n stack - tep ?	Mare & tempt = ghore -> Prest
In rep (dat * dark)? In data = dack - dap - data In data = dack - dap - not; In data * dark * dark In reptylene (duene * quene)? In reptylene (duene * quene) In duene (duene * dark In	newNade -> next = stack -> top:	
In rep (Sar * Sack) ? All tempty sort *) Solven - I've data Sol	stack -> tep = newNode;	
The Completion (Content of the Cont	3	S dhone -> Dievoi = MATT.
The Completion (Content of the Cont	int pap (stack * stack)?	to the team is burner
Sides - 1: Sides	i) (is Emoty Stack (Stack)) ?	Theo (temp).
Sid on = tack > tap = data, Stack stack (Stack); Note * tamp = stack > tap = not; Stack = tap = stack > tap = not; South (Stack); South (S	pount ("Emply Stack")	2 sielium data;
For data = stack - step = data. Note * temp = stack - tep; tack - step = stack - tep; tack (stack); post (s	Sidusin - 1;	A least beautiful beautiful
Harde * temp = clark tep; stark - tep = stark tep = rood; stark	3	t dam por
tack - ten = stack - ten - nod passil ("stack presention. "An"); ***sec (temp); **sec (temp); **sec (temp); **sec (temp)		stack stack
The (tamp); Software data; Vesta restrictuence (Quence * quence)? Quence > \$6001 = 1000C; Proposed element : \$d \land	Nede * temp = start -> top;	mil Nock (Nock);
void notificiere (airea & queue) & pt" papped element * d ln" queue — geori = Novi ; queue — geori = Novi ; nt resmptyfrieve (Queue & queue) ref ("papped element * d ln"); pop (f stack 1); ret esmptyfrieve (Queue & queue) section queue (Queue & queue) section queue (Queue & queue) f ("papped element * d ln"); pop (f stack 1); void enqueue (Queue & queue) f ("papped element * d ln"); pop (f stack 1); void enqueue (Queue & queue) f ("ueue queue f ("queue queue	stack -> tep = stack -> tep -> next.	town (stack Operations: /n");
void notificiere (airea & queue) & pt" papped element * d ln" queue — geori = Novi ; queue — geori = Novi ; nt resmptyfrieve (Queue & queue) ref ("papped element * d ln"); pop (f stack 1); ret esmptyfrieve (Queue & queue) section queue (Queue & queue) section queue (Queue & queue) f ("papped element * d ln"); pop (f stack 1); void enqueue (Queue & queue) f ("papped element * d ln"); pop (f stack 1); void enqueue (Queue & queue) f ("ueue queue f ("queue queue	taco (temp);	push (& Stock, 10);
void notificiere (airea & queue) & pt" papped element * d ln" queue — geori = Novi ; queue — geori = Novi ; nt resmptyfrieve (Queue & queue) ref ("papped element * d ln"); pop (f stack 1); ret esmptyfrieve (Queue & queue) section queue (Queue & queue) section queue (Queue & queue) f ("papped element * d ln"); pop (f stack 1); void enqueue (Queue & queue) f ("papped element * d ln"); pop (f stack 1); void enqueue (Queue & queue) f ("ueue queue f ("queue queue	option data;	pub (Astack, 20).
queue - gont = while; queue - gont = while; not resomptiqueue (Queue * queue); per((* peped dement : d \no"); per((* peped		Push C& Alack 307;
apene - 450rd = 150°C; apene - 450rd = 150°C; Int re-mpty givene (Queue * queue) solven apene (Queue * queue) 4 (Empty givene (Queue) 5 (Queue queue) 6 (Queue queue) 6 (Queue queue) 6 (Queue queue) 6 (Queue) 6 (Queue queue) 6 (Queue) 7 (Que	void intQueue Queue * queue) ?	pt " papped element . d /n"
gieve > 91091 > MULL; Pt papad alement : A th papad plement : A thought paped pleme : A thought paped plement : A thought		pep({ stack)};
Ind resimply givene (Quene * quene) Section quene (Secrit = NULL; Section quene (Mene * quene; int data)? Ment emphane (Mene * quene; int data)? Ment emphane (Mene * quene; int data)? Ment emphane (Quene) Multiply (Quene) The pepped dement is different in the peptident	alleve -> 910091 - KIULL.	pt' poped element: " d In"
verd enginene (Viene * quene; not dan)? The quene (fane); 1) (is my tylinene (quene))? 2) (is my tylinene)? 2) (quene) (quene); 3) (quene) (quene); 4) (quene) (quene) (quene) (que	2 Company of the Comp	pep (& stack)); _
verd enginene (Viene * quene; not dan)? The quene (fane); 1) (is my tylinene (quene))? 2) (is my tylinene)? 2) (quene) (quene); 3) (quene) (quene); 4) (quene) (quene) (quene) (que	int resonationique (Ougue * augue)	of "popped element: "/d /n"
verd enginene (Viene * quene; not dan)? The quene (fane); 1) (is my tylinene (quene))? 2) (is my tylinene)? 2) (quene) (quene); 3) (quene) (quene); 4) (quene) (quene) (quene) (que	autuan augue -> 18 ant == NULL:	pop(& stack));
verd enginene (Viene * quene; not dan)? The quene (fane); 1) (is my tylinene (quene))? 2) (is my tylinene)? 2) (quene) (quene); 3) (quene) (quene); 4) (quene) (quene) (quene) (que	2 = otal - otal A series	
Abde * monte de = (100 en de 1000); Inst. que le dipene ; (n"); 4) (és mysylvane que) ? que = 100 en enque (que) (p); = noundoe; enque (que , 20);	wed sometime (Along & allone : set data) &	Wielle Ollette:
1) (is Emplishere (queue)) ? queue > geont = queue > seon omprese (queue, 10); = nountate; emqueue (queue, 20);	Void englane (page * apere) to da (data).	not annie (lanone).
queue = geont = queue = seas emqueue (queue , 20);	TATIE * TENEROR = THOUGHT COM	a (" divine constitions : In").
drone = Row = drone sea author (drone 20);	1) (ormbiding (diens) 1	to character in .
= newhore; employe 14 queite, 2017;	arions - Roug = Orions - Shoul	
	= nowNexe;	
queue -> 910091 = neuntage; majeure (4 gliene)	queue -> 910091 = newNode;	emotiene (fgiene, 30);



Write A Program to Implement doubly link list with primitive operations

- a) Create a doubly linked list.
- b) Insert a new node to the left of the node.
- c) Delete the node based on a specific value
- d) Display the contents of the list

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
  int data;
  struct Node* prev;
  struct Node* next;
};
struct Node* createNode(int data) {
  struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  newNode->data = data;
  newNode->prev = NULL;
  newNode->next = NULL;
  return newNode;
}
void insertAtBeginning(struct Node** head, int data) {
  struct Node* newNode = createNode(data);
  if (*head == NULL) {
    *head = newNode;
  } else {
    newNode->next = *head;
    (*head)->prev = newNode;
    *head = newNode;
  }
}
void insertAtPosition(struct Node** head, int data, int position) {
  if (position < 1) {
    printf("Invalid position!\n");
    return;
  }
  struct Node* newNode = createNode(data);
  if (position == 1) {
    insertAtBeginning(head, data);
```

```
return;
  }
  struct Node* temp = *head;
  for (int i = 1; temp != NULL && i < position - 1; i++) {
    temp = temp -> next;
  }
  if (temp == NULL) {
    printf("Position out of bounds!\n");
    free(newNode);
    return;
  }
  newNode->next = temp->next;
  newNode->prev = temp;
  if (temp->next != NULL) {
    temp->next->prev = newNode;
  }
  temp->next = newNode;
void insertAtEnd(struct Node** head, int data) {
  struct Node* newNode = createNode(data);
  if (*head == NULL) {
    *head = newNode;
    return;
  }
  struct Node* temp = *head;
  while (temp->next != NULL) {
    temp = temp -> next;
  temp->next = newNode;
  newNode->prev = temp;
void displayList(struct Node* head) {
  if (head == NULL) {
    printf("List is empty!\n");
    return;
  struct Node* temp = head;
  printf("List contents: ");
  while (temp != NULL) {
    printf("%d ", temp->data);
```

```
temp = temp -> next;
  }
  printf("\n");
int main() {
  struct Node* head = NULL;
  int choice, data, position;
  while (1) {
     printf("\nDoubly Linked List Operations:\n");
     printf("1. Insert at Beginning\n");
     printf("2. Insert at Position\n");
     printf("3. Insert at End\n");
     printf("4. Display List\n");
     printf("5. Exit\n");
     printf("Enter your choice: ");
     scanf("%d", &choice);
     switch (choice) {
       case 1:
          printf("Enter data to insert at beginning: ");
          scanf("%d", &data);
          insertAtBeginning(&head, data);
          break;
       case 2:
          printf("Enter data to insert: ");
          scanf("%d", &data);
          printf("Enter position: ");
          scanf("%d", &position);
          insertAtPosition(&head, data, position);
          break;
       case 3:
          printf("Enter data to insert at end: ");
          scanf("%d", &data);
          insertAtEnd(&head, data);
          break:
       case 4:
          displayList(head);
          break;
       case 5:
          printf("Exiting program.\n");
          exit(0);
       default:
          printf("Invalid choice! Please try again.\n");
  }
```

```
return 0;
```

```
Doubly Linked List Operations:

    Insert at Beginning
    Insert at Position

3. Insert at End
4. Display List
5. Exit
Enter your choice: 1
Enter data to insert at beginning: 1
Doubly Linked List Operations:

    Insert at Beginning
    Insert at Position

3. Insert at End
4. Display List
5. Exit
Enter your choice: 2
Enter data to insert: 2
Enter position: 2
Doubly Linked List Operations:
1. Insert at Beginning

    Insert at Beginning
    Insert at Position
    Insert at End

4. Display List
5. Exit
Enter your choice: 3
Enter data to insert at end: 3
```

```
Doubly Linked List Operations:

1. Insert at Beginning

2. Insert at Position

3. Insert at End

4. Display List

5. Exit
Enter your choice: 4
List contents: 1 2 3

Doubly Linked List Operations:

1. Insert at Beginning

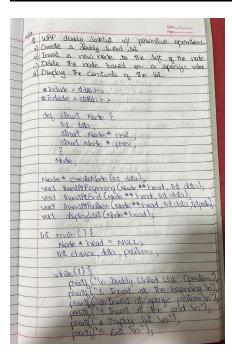
2. Insert at Position

3. Insert at End

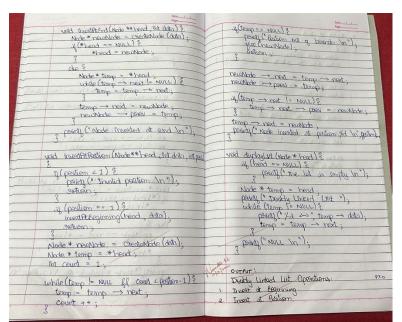
4. Display List

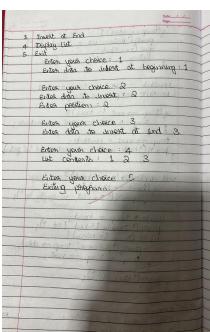
5. Exit
Enter your choice: 5
```

Exiting program.



Date	
The state of the s	default : 100 mg
parinty ("Sition years Charles);	Carlott (" T L.
pounty ("Sidest years choice");	3 tombe mono choice in 13
switch (chesce) &	3 party ("Invalid drake In");
	Siduon O:
case! (" cotra data ").	3
coset ("xd", fdata);	FLANTE OF HOLDE
mostAttegining (Chead, data).	Node * consiteNode (int dati) ?
beleak ;	Node * new Made = (Node *) mallor (speg (Node)).
2 about trival gal	
case 2:	pount ("Homery allecation billed In");
saints (" Enton data"):	ext(1): Samonan Buse and
eran ("o/d" edata);	3
gran ("/d", Adota); point ("Enter the pertion");	neuallade -> data = data :
econ (" /d", (partion);	newNode -> next = NULL;
event At Position (shear, data position)	nomphyle -> belon = Mnr1;
Laurab . Tall of of All and the traffic	siduem neutlide;
The state of the s	3
cose 3: 120 Kind Lettlewa Com	\$ (\$ \simple \
posinty ("enter data at and");	void investAt. Beginning (Node ** head, int data)?
scan ("/d" (data);	Node * nouNode = caleate Node (data);
investation (thead, data);	if (*hend == NULL) }
Prisothtend (Anena, data,	*herd = neighbode:
baeak; Shart & shift	3 like the December 114 to the
- 1300A = 1008T * 9F04	obe 9.
CONE A siling play mach tis	manufactor of the thought
diplay set (head);	(*fend) -> priev = newblade;
barok:	*1 = 1 = manha
a to the test of t	? Tena - Tenance,
COSE S: 10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	point ("Node meeted at beginning to
pointy ("Exting peregrams in").	boingto Made imposted or relational in
exit(o):	Change in the last a state of
Courter)	tolla- part for
Capt tot intruct is your	FF to the second





Write a program

- a) To construct a binary Search tree.
- b) To traverse the tree using all the methods i.e., in-order, preorder and post order
- c) To display the elements in the tree.

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

struct Node {
   int data;
   struct Node* left;
   struct Node* right;
};

struct Node* createNode(int data) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = data;
   newNode->left = NULL;
   newNode->right = NULL;
   return newNode;
}

struct Node* insert(struct Node* root, int data) {
   if (root == NULL) {
```

```
return createNode(data);
  if (data < root->data) {
     root->left = insert(root->left, data);
  } else if (data > root->data) {
     root->right = insert(root->right, data);
  return root;
void inOrder(struct Node* root) {
  if (root != NULL) {
     inOrder(root->left);
     printf("%d ", root->data);
     inOrder(root->right);
  }
}
void preOrder(struct Node* root) {
  if (root != NULL) {
     printf("%d ", root->data);
    preOrder(root->left);
    preOrder(root->right);
}
void postOrder(struct Node* root) {
  if (root != NULL) {
     postOrder(root->left);
    postOrder(root->right);
    printf("%d ", root->data);
int main() {
  struct Node* root = NULL;
  int n, value;
  printf("Enter the number of elements to insert in the BST: ");
  scanf("%d", &n);
  printf("Enter %d elements:\n", n);
  for (int i = 0; i < n; i++) {
     scanf("%d", &value);
    root = insert(root, value);
```

```
printf("\nIn-order Traversal: ");
inOrder(root);

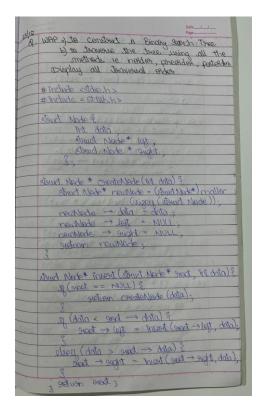
printf("\nPre-order Traversal: ");
preOrder(root);

printf("\nPost-order Traversal: ");
postOrder(root);

return 0;
}
```

```
Enter the number of elements to insert in the BST: 5
Enter 5 elements:
12 23 45 65 3

In-order Traversal: 3 12 23 45 65
Pre-order Traversal: 12 3 23 45 65
Post-order Traversal: 3 65 45 23 12
```



	polity (Was
P (tode * shear touch) reches for boar	(** che y'd demont; 'h, 'n) # (int i=0; te n; i+1) #
	107 Link 1=0; 1< n; 1++) 9
g (seed some (seet) left);	san (% d", gralue);
paint (" *d" and -> data).	3 SIRCE = TORENT (SIRCET, VIOLE);
paint ("A" enad data). marker (enat - sught);	
3 dames delentar	TO TO I TO I
	pounts (" In Inforder Tonversal: ");
1	musicales)
York payorandon (Januat Nank * arad) ?	THE PART TO A STATE OF THE PART OF THE PAR
n (spot = NULL) &	pounts (" by Rie Opider Terrusous): ");
n (seed = NOW)? point ("/d", seet - data), point idea (seet - left);	raichadox (and);
poseDisdon (and >left),	- 1 (1) Odo da Ballo I II
(there come) reshardare	pounts (" In Post Ooder Tainvarial: ");
	perthence (seed);
f had a second to be a second	
Sollin Madagetta Company of Madagette	sietunio ;
yord pathandor (struct Norte * smoot)?	3
d (anost 1 = MULL) ?	The same of the sa
pest Caden (seek -> dept).	estput
institution for the state of th	Enton the number of dements to invest in the 857:5
perint ("%d", sed > data);	in the BST 5
3	20 15 05 2
	Enter 5 dements: 12 23 45 65 3
forth it don't get but I son the but	
int main () &	Interden Traversal: 3 12 23 45 65
Start Nede * Spot = NULL.	0 0 An Tenneral : 12 3 05 45 65
Rat n Value :	Red Order Teroversal 3 65 45 23 12
to the second se	
perint ("Inter the up dements to livest	
evan ("/d", fo);	
(no/1nV)	

Program 9a

Write a program to traverse a graph using BFS method.

```
Code:
```

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
struct Queue {
  int items[MAX];
  int front, rear;
};
void initQueue(struct Queue* q) {
  q->front = -1;
  q->rear = -1;
}
int isEmpty(struct Queue* q) {
  return q->front == -1;
void enqueue(struct Queue* q, int value) {
  if (q->rear == MAX - 1) {
    printf("Queue is full\n");
    return;
  }
  if (q->front == -1) {
     q->front = 0;
  q->rear++;
  q->items[q->rear] = value;
int dequeue(struct Queue* q) {
  if (isEmpty(q)) {
    printf("Queue is empty\n");
    return -1;
  int item = q->items[q->front];
  q->front++;
  if (q->front > q->rear) {
    q->front = q->rear = -1;
```

```
return item;
void BFS(int graph[MAX][MAX], int n, int startVertex) {
  int visited[MAX] = \{0\};
  struct Queue q;
  initQueue(&q);
  visited[startVertex] = 1;
  enqueue(&q, startVertex);
  printf("BFS Traversal: ");
  while (!isEmpty(&q)) {
     int currentVertex = dequeue(&q);
    printf("%d ", currentVertex);
     for (int i = 0; i < n; i++) {
       if (graph[currentVertex][i] == 1 && !visited[i]) {
          visited[i] = 1;
          enqueue(&q, i);
  }
  printf("\n");
int main() {
  int graph[MAX][MAX], n, startVertex;
  printf("Enter the number of vertices in the graph: ");
  scanf("%d", &n);
  printf("Enter the adjacency matrix of the graph:\n");
  for (int i = 0; i < n; i++) {
     for (int j = 0; j < n; j++) {
       scanf("%d", &graph[i][j]);
     }
  }
  printf("Enter the starting vertex (0 to %d): ", n - 1);
  scanf("%d", &startVertex);
  BFS(graph, n, startVertex);
  return 0;
```

```
Enter the number of vertices in the graph: 4
Enter the adjacency matrix of the graph:
0 1 0 0
1 0 1 1
0 1 0
Enter the starting vertex (0 to 3): 0
BFS Traversal: 0 1 2 3
```

```
In dequence (stand Quence + a) ?

4) (seemely (a)) ?

5 earny (" all years is somply in ");

3 earner - 1
         278 grien Apart in graph as 38 morted
                       # include < she h>
# include < shlik h>
# define MAX 100
                                                                                                                                                                                                                                                                                                              fat Jann - q - Janus (q - Kant);

y (q - Kant - q - Janus ();

q - Kant - q - Janus - 1;
               Start Queue & Int Home Man J. Int Home J. Heren J. Heren
                                                                                                                                                                                                                                                                                                              3 stoluous thom;
       0 \rightarrow \text{gent} = -1;

0 \rightarrow \text{gent} = -1;

0 \rightarrow \text{gent} = -1;
                                                                                                                                                                                                                                                                                                       word RFS (int graph [HAX] MAX), Pit in int status)
                                                                                                                                                                                                                                                                                                                 int visited (NHC) = EOZ,
Sout Space (4);
int Quare (4);
          3 sections of a sent -
                                                                                                                                                                                                                                                                                                                Visited (stret/vestex) = 1;
emqueue (sq., stret/vestex);
pourty (" RFS Toronoval ");
       LEST ENGINE ( COUNTY OF THE VALUE ) & LEST OF THE COUNTY O
                                                                                                                                                                                                                                                                                                                while (!ustraty (fg)) & dequeue (fg);
paints ("/d", cuseum kedex);
 49 (int i = 0; i cm (++) ?

11 (angh [consententext]i] -= 1.

wated | 1 = 1.

oraque (a, 1);
a -> steams a -> steam = value.
```

```
Point (" in ");
   int graph [HAX] [HAX], n, start Vertex;
       point ("Enter the no of westices in the god"
        scan ( %d , fn);
       promit ("sites the afficency matrix to"),

prof (int 1=0; icn; i+1)?

graf (int 3=0; icn; j+1)?

scanf ("x+1, graph [] [j]]);
       point ("Enter the streeting vertex (0 to %d)"
       ecany (" of d", & stantilontex).
       BFS (geoph, n, street Vertex);
       stetuen O;
tuctua
  Enter the no of vertices in the graph 4
  Sites the adjacency matrix
  0100
 1011
 0101
 0110
Enter the starting yester (0 to 3) 0 O SES traversal: 01.8
```

Program 9b

Write a program to traverse through a graph using DFS method.

```
Code:
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int graph[MAX][MAX];
int visited[MAX];
void DFS(int vertex, int n) {
  printf("%d", vertex);
  visited[vertex] = 1;
  for (int i = 0; i < n; i++) {
     if (graph[vertex][i] == 1 && !visited[i]) {
       DFS(i, n);
     }
  }
int main() {
  int n, startVertex;
  printf("Enter the number of vertices in the graph: ");
  scanf("%d", &n);
  printf("Enter the adjacency matrix of the graph:\n");
  for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
       scanf("%d", &graph[i][j]);
     }
  }
  printf("Enter the starting vertex (0 to %d): ", n - 1);
  scanf("%d", &startVertex);
  for (int i = 0; i < n; i++) {
     visited[i] = 0;
  }
  printf("DFS Traversal: ");
  DFS(startVertex, n);
```

```
return 0;

Enter the number of vertices in the graph: 5
Enter the adjacency matrix of the graph:
0 1 1 0 0
1 0 1 1 0
1 1 0 1 1
0 1 1 0 1
0 1 1 0 1
Enter the starting vertex (0 to 4): 0
DFS Traversal: 0 1 2 3 4
```

printf("\n");

```
STO prime April a survivort at 9AW D
                                                                county ("enter the starting yestex (o to old).
                                                                can (" /d ", fotastlestex)
  #include <stdip.h>
#include <stdip.h>
                                                                yes lint i = 0; i < n; l++1 & soubcit to yes ted li] = 0; an xall grupos to
  # define MAX 100
  int visited [MAX] [MAX];
                                                                                        XAM [XAM] (hory)
                                                               paint (" DFS transacal ") gal hations
DFS ( stant Visita on );
point (" \n" ); this solver too 270
 world DFS (int vortex, int, n) }
     parinty (" !d " vortex).
       for (nt &=0, i2n; i++) & if (graph [vertex][i] == 1 ff [vertex][i]
                                                                neutore;
                    DFS(in):
                                                                Enter the no of vertices in the graph:5
                                                                Enter the adjacomy matrix
                                                                01100
                                                                10110
 int marn () &
    int n, start Vestex;
possible " Entex the no of vestires in graph");
scan; (" o/d ", fn);
                                                                11011
                                                                01101
                                                                00110
                                                                Enter starting vertex (0 to 4): 0
     printly (" Entor the adjacency matrix In")
                                                                DFS travered: 01234
     18 (3nt i = 0; ixn; i++) &
           Jes (int i = 0; jen; j++) & econ ("/d", fgeoph [i][j]);
```