



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No

Experiment - 1.

→ write a C program to display size of int, pointer and long.

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

```
int main()
{
    int a;
    int * Ptr;
    long b;
    printf ("Size of int : %.lu bytes\n", sizeof (a));
    printf ("Size of pointer : %.lu bytes\n", sizeof (ptr));
    printf ("Size of long : %.lu bytes\n", sizeof (b));
    return 0;
}
```

Output

Size of int : 4 bytes

Size of pointer : 8 Bytes

Size of long : 8 Bytes.

S
26/11/25



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

NATIONAL

Name Class Sem S.No.

Experiment - 2

im: write a C program to print 3×3 matrix using array and its transpose.

#include <stdio.h>

int main()

{

int mat[3][3];

int i, j;

11 taking input of element from the user.

Bunif ("Enter elements of 3×3 Matrix : \n")

for (i=0; i<3; i++)

{

for (j=0; j<3; j++)

{

print ("Enter element [i][j] : ", i+1, j+1);

Scanf ("%d", &mat[i][j]);

}

}

displaying the matrix

Bunif ("In the 3×3 Matrix is : \n");

for (i=0; i<3; i++)

{

for (j=0; j<3; j++)

{

print (" %d ", mat[i][j]);

}

Bunif ("\n"); → return 0;

Enter elements of 3×3 Matrix :

Enter elements [1][1] : 1
 Enter elements [1][2] : 2
 Enter elements [1][3] : 3
 Enter elements [2][1] : 4
 Enter elements [2][2] : 5
 Enter elements [2][3] : 6
 Enter elements [3][1] : 7
 Enter elements [3][2] : 8
 Enter elements [3][3] : 9

The 3×3 Matrix is :

1 2 3

4 5 6

7 8 9



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name

Class

Sem

S.No

```
#include <stdio.h>
int main()
{
    int matrix[3][3], transpose[3][3];
    int i, j;
    // Input elements of 3x3 Matrix
    printf("Enter elements of 3x3 Matrix\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("Enter element [%d][%d] : ", i + 1, j + 1);
            scanf("%d", &matrix[i][j]);
        }
    }
    // Output transpose of 3x3 Matrix
    printf("Transpose of 3x3 Matrix : \n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            transpose[j][i] = matrix[i][j];
        }
    }
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("%d ", transpose[i][j]);
        }
        printf("\n");
    }
}
```

// display the original matrix
 Using ("In original 3×3 Matrix : In "),
 for (i = 0; i < 3; i++)
 {

for (j = 0; j < 3; j++)

printf ("%d ", matrix[i][j]),

printf ("\n"),

O/P

Enter elements of 3×3 Matrix;

1 2 3
4 5 6
7 8 9

Original 3×3 Matrix.

1 2 3
4 5 6
7 8 9

Transport of the Matrix

for ($j = 0$; $j < 3$; $j++$)

Print ($"\n".$ $\text{d} \& \text{ t}"$, transpose [i][j]);

Print ($"\n"$);

Matrix;

Q
of 11/25



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

ONAL

Name

Class

Sem S.No.

Output :

```

Stack Menu .
1) Push
2) Pop
3) Peek
4) display
5) Exit

Enter your choice : 1
Enter value to push 30;
30, 20, 10 pushed into stack

Stack Menu .
1) Push
2) Pop
3) Peek
4) display
5) Exit

Enter your choice : 2
30 popped from Stack

```

INSTITUTE OF ENGINEERING AND TECHNOLOGY
 (Mohansingh Sukhadia University, Udaipur)



Name Class Sem S.No.

Program = 4

#include < stdio.h >
 #include < conio.h >
 #define size 5 // Maximum size of the stack.

```

int stack [size];
int top = -1; // Initially stack is empty
  
```

// Function to push an element into the stack

```

void push ( int value )
  
```

```

if ( top == size - 1 )
  printf ("Stack Overflow! Cannot push %d\n",
  value );
else {
  top++;
  
```

```

  stack [top] = value;
  
```

```

  printf ("%d pushed into stack, %n", value );
  
```

```

}
  
```

```

// Functional to pop an element from the stack
void pop ( )
  
```

```

if ( top == -1 )
  printf ("Stack Underflow! Stack is empty\n");
  
```

```

}
  
```

Output

Stack menu
1. Push
2. Pop
3. Peek
4. display

5. exist

Enter your choice : 4

Stack element

20

10

Stack menu.

1. Push
2. Pop
3. Peek
4. display
5. exist

Enter your choice : 7

Invalid choice

Stack menu

1. Push

Pop

Peek

display

exist

Enter your choice : 5

INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Mohansingh Sukhadia University, Udaipur)
Name Class Sem S.No.

3
else {
 if (top == -1)
 printf ("Stack is empty! No top element\n");
 else {
 printf ("Top element is: %d\n", stack[top]);
 }
}
// Function to display ()
if (top == -1)
 printf ("Stack is empty\n");
else {
 printf ("Stack elements are: \n");
 for (i = top; i >= 0; i--)
 printf ("%d → value %d\n", i, stack[i]);
}
3
int main () {
 int choice, value;
 choice();
}



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

ONAL

Name Class Sem S.No.

```
printf ("Simple Stack Program using Array\n");
while (1)
{
```

```
    printf ("\n --- Stack Menu -- \n");
    printf ("\n -1 Push In 2 , Pop In 3 , Peek In 4 ,
            Display In 5 Exit n"),
    scanf ("%d" & choice);
    switch (choice){
```

Case 1;

```
        printf ("Enter value to - push : ");
        scanf ("%d", & value);
        push (value);
        break;
```

Case 2;

```
        pop ();
        break;
```

Case 3;

```
        peek ();
        break;
```

Case 4 . id

```
        display ();
        break;
```

Case 5 :

```
        return 0;
```

Default :

```
        printf (" Invalid choice ! Please try again
                'In' ) );
```

??

?

O/P

----- Queue Menu -----

- 1) Enqueue
- 2) Dequeue
- 3) Peep

- 4) Display

- 5) Exit

Enter your choice : 1
Enter value to insert : 3
3 inserted into queue.



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name

Class

S.No

Program - 5

Aim: WAP to display the equation of queue.

#include < stdio.h >

#include < conio.h >

define Max

int queue[Max];

int front = -1;

int rear = -1;

void enqueue (int x)

{ if (rear == Max - 1)

 printf ("Queue is full !\n");

else

{ if (front == -1)

 front = 0;

 rear++;

 queue [rear] = x;

 printf ("1. d inserted into queue [n, x];

 void dequeue () {

 if (front == -1 || front > rear)

 {

 printf ("Queue is empty !n");



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

else

{

printf ("%d removed from queue\n", queue[front]);
front++;
}

void peek ()

{

if (front == -1 || front > rear)
{

printf ("Queue is empty\n");
}

else

{

printf ("Front element : %d\n", queue[front]);
}

void display ()

{

if (front == -1 || front > rear)
{

printf ("Queue is empty\n");
}

else

{

printf ("Queue elements : ");

for (int i = front; i <= rear; i++)
{

{

printf ("%d", queue[i]);

}



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

printf ("1n");
scanf ("%d",

int main()
{

int choice value;
while (1)
{

printf ("1n --- Queue Menu --- \n");

printf (" 1, Enqueue \n");

printf (" 2 Dequeue \n");

printf (" 3 Peek \n");

printf (" 4 Display \n");

printf (" 5 Exit \n");

printf (" Enter your choice : ");

Scanf ("%d", &choice);

switch (choice)

{

case 1:

printf (" Enter value to insert : ");

scanf ("%d %d", &d, &value);

enqueue (value);

break;

case 2 :

dequeue ();

break;

case 3 :

peek ();

break;



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

case 4:

display ();
break;

case 5 :

return 0;
default :

printf (" Invalid choice In ");

3

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

O/P



Name Class Sem S.No.

Program - 6

Ques : Map the display basic operation of circular queue.

- 1) Enqueue
 - 2) Dequeue
 - 3) Peek
 - 4) Display
 - 5) Exit
- Enter your choice : 1
 Enter value to insert : 12
 12 inserted

```
#include < stdio.h>
#define SIZE 10 // size of circular queue.
int queue[SIZE], front = -1, rear = -1;
// function to insert element
void enqueue (int value)
{
    if ((front == -1) && (rear == (SIZE - 1))) // rear + 1 == front
        printf ("Queue is full : In ");
    else
        {
            if (front == -1) // first element
                front = 0;
            rear = (rear + 1) % SIZE;
            printf ("Enqueue : Value ");
            printf (" %d inserted /n ", value);
        }
}
// function to delete element
```



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

void dequeue ()

{

if (front == -1)
else

printf ("Queue is Empty : In");
else

{

printf ("%.d deleted In", queue [front]);
if (front == rear)
else

// Queue become empty

front = rear = -1;
else

{

front = (front + 1) % size ;
}

3

3

// function to show front element.

void peek()

{

if (front == -1)
else

printf ("Queue is EMPTY / In");
else

else

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

printf ("front element : %d \n", queue[front]);

// function to display queue
void display()

if (front == -1)

printf ("Queue is EMPTY ! \n");

else

printf ("Queue : ");
int i = front;
while (i)

printf ("%d ", queue[i]),
if (i == rear)

break;

i = (i + 1) % rear;

printf ("\n");

// Main function

int main()



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

int choice, value;
while (1)

{

printf ("1. Enqueue\n2. Dequeue\n3. Peek\n4. Display\n5. Exit\n");

printf ("Enter your choice : ");

scanf ("%d", &choice);

switch (choice)

{

case 1:

printf ("Enter value to insert : ");

scanf ("%d", &value);

enqueue (value);

break;

case 2:

dequeue ();

break;

case 3:

peek ();

break;

case 4:

display ();

break;

case 5:

default:

printf ("Invalid choice (\n");

}

};

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohandas Sukhadia University, Udaipur)



Name

Class

Sem

S.No.

total number of elements : 5
Enter the element : 4
1 5 3 4 2
Search position : 3
Element found at
Element found at

total number of elements : 5
Enter the element : 4
1 5 3 4 2
Search position : 3
Element found at
Element found at

Aim = WAP to explain linear search operator.
#include < stdio.h >
int main()
{

int a[50], n, i, key; found = 0;
printf ("Enter number of elements : ");
scanf ("%d", &n);
printf ("Enter the elements : ");
for (i=0; i<n; i++)
{

scanf ("%d %c", &a[i]);

printf ("Enter element to search : ");
scanf ("%d", &key);
for (i=0; i<n; i++)

{
if (a[i] == key)
{
found = 1;
break;
}

if (found == 1)
printf ("Element found at position %d", i+1);
else
printf ("Element not found");

3.3

2.5 (found == 0)
3. Helium O;

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name

Class

Sem

S.No.

Program 8

Aim: write a program to explain linear Search operation.

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

```
int main()
```

```
{
```

```
int a[50], n, i, Key found = 0;
```

```
printf ("Enter number of element : ");
```

```
scanf ("%d", &n);
```

```
printf ("Enter the element : In ");
```

```
for (i=0, i < n; i++)
```

```
{
```

```
scanf ("%d", &a[i]);
```

```
printf ("Enter element to Search : ");
```

```
scanf ("%d", &Key);
```

```
for (i=0; i < n; i++)
```

```
{
```

```
if (a[i] == Key)
```

```
{
```

```
printf ("Element found at position %d, %d, i+1);
```

```
found = 1;
```

```
break;
```

```
}
```

```
} if (found == 0)
```

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohansingh Sukhadia University, Udaipur)

Name Class Sem S.No.

Output →

Enter number of element : 5

Enter the elements : /

1 5 8 11 4

Enter element to search : 8

Element found at position : 3

Found (" Element not found "),
Sukhadia

3

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanalal Sukhadia University, Udaipur)

Name Class Sem S.No.

Experiment - 9

write a program to display Binary Search operation on a array.

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

```
int main()
```

```
{
```

```
int a[50], n, i, Key, low, high, mid; found = 0;
```

```
printf ("Enter number of elements : ");
```

```
scanf ("%d" & n);
```

```
printf ("Enter elements in sorted order : ");
```

```
for (i = 0; i < n; i++)
```

```
{
```

```
scanf ("%d" & a[i]);
```

```
}
```

```
printf ("Enter element to search : ");
```

```
scanf ("%d" & Key);
```

```
low = 0;
```

```
high = n - 1;
```

```
while (low <= high)
```

```
{
```

```
mid = (low + high) / 2;
```

```
if (a[mid] == Key)
```

```
{
```

```
printf ("Element found at position %d",
```

```
mid + 1);
```

```
found = 1;
```

Enter number of elements : 6
Enter elements in sorted order :

1 3 5 7 9 11

Enter element to Search : 9

Element found at position 5

INSTITUTE OF ENGINEERING AND TECH

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem

break;

}

else if (a[mid] < Key)

{

low = mid + 1;

}

else

{

high = mid - 1;

}

if (found == 0)

{

printf ("Element not found");

return 0;

}

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

Experiment - 10

Aim: write a program to reverse a list using stack

```
#include <stdio.h>
#define MAX 100 // maximum size of stack
int stack[MAX];
int top = -1;
// Function to push element onto stack
void push(int value) {
    if (top == MAX - 1)
        printf("Stack Overflow\n");
    else
```

```
        stack[++top] = value;
```

```
int main() {
```

```
    int list[MAX], n, i;
```

```
    printf("Enter number of elements: ");
    scanf("%d", &n);
```

```
    printf("Enter elements: \n");
```

```
    for (i=0; i<n; i++)
```

```
        scanf("%d", &list[i]);
```

```
// Push all elements into stack
```

```
for (i=0; i<n; i++)
    push(list[i]);
```

```
// Pop all elements back into list (reversed)
```

```
for (i=0; i<n; i++)
    list[i] = pop();
```

Enter number of element : 5

Enter element :

1 4 7 8 9

Reversed list : 9 8 7 4 1



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name

Class

Sem

S.No.

// Function to pop element from Stack
int pop () {

if (top == -1) {

printf ("Stack Underflow");

return -1;

else

return stack [top--];

}

printf ("In Reversed list:");

for (i = 0; i < n; i++)

printf ("%d"; list[i]);

return 0;

}

int pop () {

if (top == -1)

return -1; // Stack underflow.

else

return stack [top--];

}

int main () {

int n, i;

long long fact = 1;

printf ("Enter a number : ");

scanf ("%d", &n);

Enter a number : 4

Factorial of 4 = 24.

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

Experiment - 511

Aim: write a program to calculate factorial of a number using stack.

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

```
#define MAX 50
```

```
int stack[MAX];
```

```
int top = -1;
```

```
void push(int n) {
```

```
    if (top == MAX - 1)
```

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

```
    else,
```

```
        stack[++top] = n;
```

```
}
```

```
// Push numbers 1 to n
```

```
for (i=1; i<=n; i++) {
```

```
    push(i);
```

```
}
```

```
// Pop and multiply
```

```
while (top != -1) {
```

```
    fact = fact * pop();
```

```
}
```

```
printf("Factorial of %d = %ld\n", n, fact);
```

```
return 0;
```

```
}
```

Enter number of terms in the polynomial: 3
Enter coefficient / power of x and power of y
Enter coefficient / power of x and power of y

5	2	1
3	2	1
2	1	0

Polynomial Representation
Coefficient X-power Y-power

5	2	1
3	2	1
2	1	0

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

Experiment : 12

write a program to represent a 2 variable Polynomial using array.

```
#include <stdio.h>
int main()
{
    int poly[20][3];
    int n, i;
    printf("Enter number of terms in the polynomial");
    scanf("%d", &n);
    printf("Enter coefficient power of x and power of y: \n");
    for (i = 0; i < n; i++)
    {
        scanf("%d %d %d", &poly[i][0], &poly[i][1], &poly[i][2]);
        // poly[i][0]          // Coefficient
        // poly[i][1]          // power of x
        // poly[i][2]          // power of y.
    }
    printf("In Polynomial Representation : \n");
    printf("Coefficient X-power Y-power \n");
    for (i = 0; i < n; i++)
    {
        printf("%d %d %d \n", poly[i][0], poly[i][1], poly[i][2]);
    }
    return 0;
}
```

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

Experiment - 13

write a program to represent Sparse Matrix using Array.

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

```
int main () {
```

```
    int r, c, i, j;
```

```
    int mat[10][10];
```

```
    int Sparse[50][3];
```

```
    int K = 1
```

```
    printf ("Enter number of rows and columns : ");
```

```
    scanf ("%d %d", &r, &c);
```

```
    printf ("Enter element of the matrix : \n");
```

```
    for (i = 0; i < r; i++) {
```

```
        for (j = 0; j < c; j++) {
```

```
            scanf ("%d", &mat[i][j]);
```

```
}
```

```
/* First row stores rows, columns and non-Zero  
count */
```

```
Sparse[0][0] = r;
```

```
Sparse[0][1] = c;
```

```
Sparse[0][2] = K;
```

Enter number of rows and columns : 3 3

Enter elements of the matrix:

0 0 1 1 2 0 0 0 0 0 3

Sparse Matrix Representation

Row	Col	Value
3	3	3
0	2	1
1	0	2
2	2	3

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

/* Convert to Sparse form */

for (i=0; i<x; i++) {

 for (j=0; j<c; j++) {

 if (mat[i][j] != 0) {

 Sparse[K][0] = i;

 Sparse[K][1] = j;

 Sparse[K][2] = mat[i][j];

 K++;

 Sparse[0][2]++;

}

}

}

printf ("In Sparse Matrix Representation : \n ");

printf ("Row Col Value \n ");

for (i=0; i<=Sparse[0][2]; i++) {

 printf ("%d %d %d \n ",

 Sparse[i][0], Sparse[i][1], Sparse[i][2]);

}

return 0;

}

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.No.

Experiment - 14.

write a program to show the Bubble Sort Algorithm

```
#include < stdio.h>
int main ()
{
    int a[50], n, i, j, temp;
    printf ("Enter no. of element ");
    scanf ("%d", &n);
    printf ("Enter the element ");
    for (i=0, i<n, i++)
    {
        scanf ("%d", &a[i]);
    }
    for (i=0, j<n-1; j++)
    {
        if (a[j] > a[j+1])
            if (a[0] > a[1])
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
    }
    printf ("Sorted array ");
    for (i=0, i<n, i++);
}
```

Output

Enter number of elements : 5

Enter the elements : 7 3 1 4 8

Sorted array : 1 3 4 7 8

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name Class Sem S.N.

```
printf (" Allocated array ");
for (i= 0; i < n; i++)
{
```

```
    printf ("%d", a[i]);
}
```

```
return 0;
```

```
}
```

output

Enter number of elements : 5

Enter the elements : 4 8 1 9 7

Sorted array :

1 4 6 7 9

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name..... Class..... Sem..... S. No.....

Program - 15

Aim: WAP to Show the Insertion Sort Operation.

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

```
int main()
```

```
{
```

```
int a[50], n, i, j, Key;
```

```
printf ("Enter number of elements: ");
```

```
scanf ("%d", &n);
```

```
printf (" Enter the element : \n");
```

```
for (i=0; i<n; i++) {
```

```
    scanf ("%d", &a[i]);
```

```
}
```

```
// Insertion sort logic
```

```
for (i=1; i<n; i++) {
```

```
    j = i-1;
```

```
    while (j >= 0 && a[j] > key) {
```

```
        a[j+1] = a[j];
```

```
        j--;
```

```
}
```

~~a[j+1] = key;~~

~~3~~

```
    printf ("Sorted array : \n");
```

```
    for (i=0; i<n; i++) {
```

```
        printf ("%d", a[i]);
```

```
}
```

```
return 0;
```

~~3~~

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name.....

Class.....

Sem.....

S. No.....

Program 16.

WAP a pro to implement Stack using linked list.

```
#include <stdio.h>
#include <stdlib.h>
// define node structure.
- Struct Node {
    int data;
    Struct Node *next;
};

Struct Node * top = NULL;
;

Struct Node * top = NULL;
// Push operation
void push (int item) {
    Struct Node * newNode;
    newNode = (Struct Node*) malloc (sizeof(Struct
Node));
    if (newNode == NULL) {
        printf ("Stack Overflow \n");
        return;
    }
    newNode->data = item;
    newNode->next = top;
    top = newNode;
    printf ("%d pushed into Stack \n", item);
}
```



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name.....

Class.....

Sem.....

S. No.....

// Pop operation

void pop () {

struct Node * temp;

if (top == NULL) {

printf ("Stack Underflow\n");

return;

}

temp = top;

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

top = top->next;

free (temp);

}

// display stack

void display () {

struct Node * temp;

if (top == NULL) {

printf ("Stack is empty\n"),

return;

}

temp = top;

printf ("Stack elements : \n");

while (temp != NULL) {

printf ("%d ->", temp->data);

temp = temp->next;

}

printf ("\nNULL\n");

}

----- Stack using linked list -----

1. Push
2. Pop
3. Display
4. Exit

Enter your choice: 1

Enter element to push: 5

5. pushed onto Stack.

----- Stack using linked list -----

1. Push
2. Pop
3. Display
4. Exit

Enter your choice: 3



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name..... Class..... Sem..... S. No.....

11 Main function.

int choice, item ;
while (1) {

```
    printf ("1. n - stack using linked list\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 element to push : ");
    scanf ("%d", &item);
    push (item);
    break;
```

Case 2:

```
    pop ();
    break;
```

case 3:

```
    display ();
    break;
```

case 4

```
    exit (0);
```

default :

```
    printf ("Invalid choice (n),
```

3

3 return 0;

3



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name..... Class..... Sem..... S. No.....

Program 17.

WAP to implement Queue using linked list.

```
#include < stdio.h>
#include < stdlib.h>
// define node structure
Struct Node {
    int data;
    Struct Node * next;
};

Struct Node * front = NULL;
Struct Node * rear = NULL;
```

1 Enqueue operation

```
Void enqueue (int item) {
    Struct Node *newNode;
    newNode = (Struct Node *) malloc (sizeof (Struct
        Node));
    if (newNode == NULL) {
        printf ("Queue Overflow !\n");
        return;
    }
```

3

```
    newNode-> data = item;
    newNode-> next = NULL;
    if (rear == NULL) { "Queue is empty"
        front = rear = newNode;
    }
```

3 else {

```
    rear-> Next = newNode;
}
```



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name..... Class..... Sem..... S. No.....

3
rear = newNode;

{
printf (" % inserted into queue \n", item);

// Dequeue operation.

void dequeue () {

struct Node * temp;

if (front == NULL) {

printf (" Queue Underflow \n");

return;

3

temp = front;

printf (" Selected element : %d \n", front->data);
front = front -> next;

if (front == NULL) {

rear = NULL;

3

free (temp);

3

// Display queue

void display () {

struct Node * temp;

if (front == NULL) {

printf (" Queue is empty \n");

return;

3



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name..... Class..... Sem..... S. No.....

```
temp front;
printf ("Queue elements : (n"),
while (temp != NULL) {
    printf ("%d->", temp->data);
    temp = temp->next;
}
printf ("NULL (n);
```

1 Main function.

```
int main () {
```

```
    int choice, item;
```

```
    while (1) {
```

```
        printf ("1. Enqueue (n);
        printf ("2. Dequeue (n);
```

```
        printf ("3. Display (n);
```

```
        printf ("4. Exit (n);
```

```
        printf ("Enter your choice : ");
```

```
        scanf ("%d", &choice);
```

```
        switch (choice) {
```

```
            case 1:
```

```
                printf ("Enter element to insert : ");
                scanf ("%d", &item);
```

```
                enqueue (item);
```

```
                break;
```

```
            case 2:
```

```
                dequeue ();
```

```
                break;
```

Output

----- Queue using linked list -----

1. Enqueue
2. Dequeue
3. Display
4. Exit

Enter your Choice : 3

2 → 6 → NULL



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name.....

Class.....

Sem.....

S. No.....

case 3 :

display();

break;

Case 4,

exit(0);

default

3

prompt ("Invalid choice \n");

3

defaule(0);

3



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name..... Class..... Sem..... S. No.....

Program - 18
WAP for BFS using Adjacency Matrix.

#include < stdio.h >
int n;

int adj[20][20];

int visited[20];

int queue[20], front = -1, rear = -1;

void BFS (int start);

{

int i, v;

printf ("%d", start);

visited[0] = 1;

queue[++rear] = start;

while (front != rear)

{

v = queue[++front];

for (i = 0; i < n; i++)

{

if (adj[v][i] == 1 && visited[i] == 0),
 {

 printf ("%d", i);

 visited[i] = 1;

 queue[++rear] = i;

}

3

3

3

Enter number of vertices : 3
Enter adjacency matrix :

1	2	3
4	5	6
7	8	9

Enter starting vertex (0 to 2) ; 0
BFS Traversal : 0

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanalal Sukhadia University, Udaipur)

Name.....

Class.....

Sem.....

S. No.....

int main()

```
int i, j, start;
printf ("Enter number of vertices : ");
scanf ("%d", &n);
printf ("Enter adjacency matrix : m");
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        scanf ("%d", &adj[i][j]);
for (i=0, i<n; i++)
    visited[i]=0;
printf ("Enter starting vertex (0 to %d); ", n-1);
scanf ("%d", &start);
printf ("BFS Traversal: ");
BFS(start);
return 0;
```

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name

Class Sem S.No.

WAP to implement DFS using Adjacency Matrix
Program - 19

#include < stdio.h >

int n;

int adj [20][20];

int visited [20];

void DFS (int v)

{

int i;

printf (" %d ", v);

visited [v] = 1;

for (i=0; i<n; i++)

{

if (adj [v][i] == 1 && visited [i] == 0)

DFS (i);

}

3

3

int main()

{

int i, j, start;

printf (" Enter number of vertices : ");

scanf ("%d", &n);

printf (" Enter adjacency matrix : \n "),

Enter number of vertices : 6

Enter adjacency matrix:

1	2	3	4	5	6
1	2	3	4	5	6
3	4	5	6	7	8
4	5	6	7	8	9
3	4	6	7	8	9

2 3 4 5 6 7 8

Enter starting vertex (0 to 5) : DPS Traversal :



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name.....

Class..... Sem..... S. No.....

```
for (i=0; i<n; i++)  
    for (j=0; j<n; j++)  
        scanf ("%d", &adj[i][j]);  
    for (i=0; i<n; i++)  
        visited[i] = 0;  
    printf ("Enter starting vertex (0 to %d); ", n-1);  
    scanf ("%d", &start);  
    printf ("DFS Traversal : ");  
    DFS(start);  
    return 0;  
3
```



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name.....

Class.....

Sem.....

S. No.....

Program : 2.0.

Write a Program to show basic operation of a
Binary Tree.

```
#include <stdio.h>
#define MAX 50
int tree[MAX];
int n = 0;
/* Insert element */
void insert (int item) {
    if (n >= MAX - 1) {
        printf ("Tree is full! \n");
        return;
    }
    n++;
    tree[n] = item;
    printf ("Item inserted successfully. \n");
}

/* Delete element */
void deleteItem (int item) {
    int i, found = 0;
    for (i = 1; i <= n; i++) {
        if (tree[i] == item) {
            found = 1;
            break;
        }
    }
}
```

3

3



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name.....

Class.....

Sem.....

S. No.....

```
postorder(2 * i + 1);  
printf ("%d", tree[i]);  
3  
/* Main function */  
int main () {  
    int choice, item;  
    while (1) {  
        printf ("In -- Array-Based Binary Trees - In");  
        printf ("1. Insert (%d)");  
        printf ("2. Delete (%n)");  
        printf ("3. Inorder Traversal (%n)");  
        printf ("4 Preorder Traversal (%n)");  
        printf ("5 Postorder Traversal (%n)");  
        printf ("6 Exit (%n)");  
        printf ("Enter your choice : ");  
    }  
}
```

```
scanf ("%d", &choice);  
switch (choice) {
```

case 1 :

```
    printf ("Enter item to insert : ");  
    scanf ("%d", &item);  
    insert (item);  
    break;
```

case 2

```
    printf ("Enter item to delete : ");  
    scanf ("%d", &item);  
    deleteItem (item);  
    break;
```

--- Array Based Binary Tree ---

1. Insert
2. Delete
3. Inorder Traversal
4. Preorder Traversal
5. Postorder Traversal
6. Exit

Enter your choice : 1

Enter item to insert : 10

Item inserted successfully



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

NATIONAL

Name.....

Class.....

S. No.....

```
if (!found) {
    printf ("Item not found! \n");
    return;
}

tree[i] = tree[n]; // replace with last
element n - i
printf ("Item deleted successfully. \n");

/* Inorder Traversal */
void inorder (int i) {
    if (i > n)
        return;
    inorder (2 * i);
    printf ("%d ", tree[i]);
    inorder (2 * i + 1);
}

/* Preorder Traversal */
void preorder (int i) {
    if (i > n)
        return;
    printf ("%d ", tree[i]);
    preorder (2 * i);
    preorder (2 * i + 1);
}

/* Postorder Traversal */
void postorder (int i) {
    if (i > n)
        return;
    postorder (2 * i);
}
```



INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Mohanlal Sukhadia University, Udaipur)

Name.....

Class.....

Sem.....

S. No.....

Case 3;

```
printf ("Inorder Traversal : ");
morder (i);
printf ("In");
break;
```

Case 4;

```
printf ("Preorder Traversal : ");
preorder (i);
printf ("In");
break;
```

Case 5;

```
printf ("Postorder Traversal : ");
postorder (i);
printf ("In");
break;
```

Case 6;

selwin 0;
default;

3

printf ("Invalid choice (In),

3

P

17102126