

Certificate

Name: Anand Kumar Kushwaha

Class: 2nd year

Roll No: 19CS021

Exam No:

Institution Anand International College of Engineering

This is certified to be the bonafide work of the student in the
D.S.A Laboratory during the academic
year 20 / 20 .

No. of practicals certified _____ out of _____ in the
subject of _____

.....
Teacher In-charge

.....
Examiner's Signature

.....
Principal

Date:

Institution Rubber Stamp

(N.B. The candidate is expected to retain his/her journal till he/she passes in the subject.)

Index

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
1	Number Using Recursion				
2	Factorial Using Recursion				
3	factorial n number using Recursion				
4	optimized program in case of multiplication				
5	Fibonacci Series Using Recursion				
6	Fibonacci Series Using Iteration method				
7	Fibonacci Series using memoization method				
8	Tower of Hanoi using Recursion				
9	Recursive Linear Search problem				
10	Iterative Binary Search problem				
11	Recursive binary search problem				
12	Bubble sort technique with optimized				
13	Insertion Sort technique				
14	Selection Sort technique				
15	Quick Sort consider last element as pivot				
16	Quick Sort Considering first element as pivot				
17	Merge Sort				
18	Shell Sort				
19	Radix Sort				
20	H-Count Sort				
21	Heap Sort				
22	Singly linked list				
23	Doubly linked list				
24	Circular linked list				
25	Stack Using Array				
26	Stack using linked list				
27	queue using array				

Index

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
28	Queue Using Linked List				
29	Circular Queue Using Array				
30	Circular Queue Using Linked List				
31	Queue Using Stack			31/12/2020	
32	deque (double ended queue)				
33	Priority Queue				
34	Find inorder, preorder and postorder				
25					

Digitized by

Enter value of $n=5$
Sum of n natural numbers is $=15$.

Date: 5/1/2014
 Page No. 4

(i) Write a program to perform sum of n natural numbers using recursion.

```
#include <stdio.h>
int result();
int n;
int main()
{
    printf("Enter value of n = ");
    scanf("%d", &n);
    result = sum(n);
    printf("Sum of n natural numbers (%d) = %d", n, result);
    return 0;
}

int sum(int n)
{
    if (n == 0)
        return n;
    else
        return sum(n-1) + n;
}
```

Output

Enter value of n = 5

~~Factorial of n is 120.~~

Date: 5/5/2022

Page No. 2

Expt No. 2

2. Write a program to perform factorial of a number using recursion.

```
#include <iostream>
int main()
```

```
int fact();
cout << "Enter value of n = ";
cin >> n;
fact = Factorial(n);
cout << "Factorial of n = " << fact;
```

```
int Factorial(int n)
```

```
{ if (n <= 1)
    return 1;
else
    return n * Factorial(n - 1);}
```

Teacher's Signature:

Output

Enter value of n: 5
Fibonacci = 1134.
Enter value of m: 5
calculated power = 3125

Date 9/10/2020

Page No. 3

Page No. 3

Q3. write a program to perform factorial of a number using recursion.

```
#include < stdio.h >
int main()
{
    int n,m,pow();
    printf ("Enter value of n: ");
    scanf ("%d", &n);
    printf ("Enter value of m: ");
    scanf ("%d", &m);
    pow();
    printf ("calculated power is %d", pow);
}

int pow(int n,int m)
{
    if (m == 0)
        return 1;
    else
        return pow(n,m-1)*n;
}
```

Teacher's Signature

218

Entered values of $m = 3$
Entered values of $m = 5$
Calculated power is = 12.

Expt No.	Date _____
Page No.	Page No. _____
4	Write a program to calculate power. In case of multiple factors.
	#include <iostream>
	int main()
	{
	int n,m,pow=1;
	cout << "Enter value of n = " ;
	cin >> n ;
	cout << "Enter value of m = " ;
	cin >> m ;
	pow = pow * n ;
	pow = pow * m ;
	cout << "Calculated Power is = " << pow << endl ;
	}
	int pow(int n,int m)
	{
	if (m==0)
	return 1 ;
	else if (m>0, n==0)
	return -1 ;
	else
	return (n * pow(n,m-1)) ;
	}

Output

Enter the value of n 345
113490870.
Time taken \Rightarrow 11.37 sec.

Date: 14/11/2018	
Page No.	5
5.	Write a program to implement Fibonacci Number using recursion and also find its running time using clock + micro .
	#include <stdio.h> #include <time.h> int fib(int n); { if (n == 1) { return 1; } return fib(n-1) + fib(n-2); } int main() { int n; clock_t begin, end; double time_spent; printf ("Enter the value of n: "); scanf ("%d", & n); begin = clock(); printf ("Value: ", fib(n)); end = clock(); time_spent = (double) (end - begin) / CLK_TCK; printf ("Time taken \Rightarrow %f sec", time_spent); return 0; }

Teacher's Signature _____

Output

Enter the value of n \Rightarrow 45
1134983199
Time taken \Rightarrow 0.000000

Date 14/11/2023

Page No. 6

Q. write a program to implement Fibonacci Series using
Memorization method to optimize execution and also find
its running time using clock & memory.

```
#include <iostream>
#include <cmath>
int F[50];
int f(int n)
{
    if (n == 0)
        f[0] = 0;
    if (n == 1)
        f[1] = 1;
    if (F[n] != -1)
        return F[n];
    F[n] = f(n-1) + f(n-2);
}
int main()
{
    int n;
    cout << "Enter begin, end : ";
    double time_spent;
}
```

Teacher's Signature : _____

```
printf("Enter the value of n : ");
scanf("%d", &n);
begin = clock();
printf("%d\n", fib(n));
end = clock();
time_spent = (double)(end - begin) / CLK_TCK;
printf("Time taken for %d, time_spent");
return 0;
```

{}

Teacher's Signature : _____

Output

Enter the value of n \Rightarrow 45
1134903170
Time taken \Rightarrow 0.000000

Expt. No. 7

Date: 08/11/2020

Page No. 8

7. Write a program to implement Fibonacci Series using
Iteration method and also find its running time using
clock() macro.

```
#include <stdio.h>
#include <time.h>
int fib(int n);
int i;
int f[50];
f[0]=0;
f[1]=1;
for(i=2; i<=n; i++)
    f[i] = f[i-1] + f[i-2];
printf("%d", f[n]);
return 0;
int main();
{
    int i;
    char start, end;
    double time_spent;
    printf("Enter the value of n  $\Rightarrow$  ");
    scanf("%d", &n);
    begin = clock();
    printf("%d", fib(n));
    end = clock();
    time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
    printf("\nTime taken %f sec", time_spent);
}
```

Teacher's signature:

Date _____

Expt. No. _____

Page, No. 9

```
end = clock();  
time_spent = (double) (end - begin) / CLK_TCK;  
printf ("Time taken \rightarrow %.4f\n", time_spent);  
return 0;
```

4.

Teacher's Signature : _____

Output

Enter the number of disk > 3
 Move DISK 1 from rod A to rod B
 Move DISK 2 from rod A to rod C
 Move DISK 1 from rod B to rod C
 Move DISK 3 from rod A to rod B
 Move DISK 2 from rod C to rod A
 Move DISK 2 from rod C to rod B
 Move DISK 1 from rod A to rod B.

Expt. No.	Date	Page No.
8	24/11/2021	10
Q. Write a program to implement Tower of Hanoi problem using recursion.		
#include <iostream.h> void move_n_of_hanoi_Link n, char source_rod, char destination_rod, char aux_rod { if (n == 1) { cout << "Move disk 1 from rod " << source_rod << " to rod " << destination_rod << endl; } else { move_n_of_hanoi_Link n-1, source_rod, aux_rod, destination_rod; cout << "Move disk " << n << " from rod " << source_rod << " to rod " << destination_rod << endl; move_n_of_hanoi_Link n-1, aux_rod, destination_rod, source_rod; } }		
int main() { int n; cout << "Enter the number of disk > "; cin >> n; move_n_of_hanoi_Link n, 'A', 'B', 'C'; return 0; }		
Teacher's Signature:		

Output

Element 3 is present at index 4.

Expt. No.	Date _____	Page No.
Q		11
4. Write a program to implement recursive linear Search problem.		
#include <stdio.h> int RLS (int arr[], int l, int r, int key) { if (l > r) return -1; if (arr[l] == key) return l; if (arr[r] == key) return r; return RLS (arr, l+1, r-1, key); } int main () { int arr [] = {12, 34, 56, 7, 8}; int n = sizeof (arr) / sizeof (arr[0]); int * = 3; int index = RLS (arr, 0, n-1, *); if (index != -1) printf ("Element %d is present at index %d", *, index); else printf ("Element %d is not present", *); return 0; }		
Teacher's Signature _____		

Output

Element is present at index 3.

Expt. No.	1.2	Date	10/10/2018
		Page No.	12
10.	Write a program to implement iterative binary search problem.		
	#include <stdio.h> int search(int arr[], int n, int m) { int l; for (l = 0; l < n; l++) if (arr[l] == m) return l; return -1; } int main() { int arr[] = {2, 3, 4, 10, 40}; int n = 10; int m = search(arr, n, 10); int result = search(arr, n, m); if (result == -1) printf("Element is not present in array\n"); else printf("Element is present at index %d", result); return 0; }		
	Teacher's Signature _____		

Output

Element is present at index 3

Expt No.	Page No.
11	15
Ques. Write a program to implement Standard Binary Search.	
Ans.	
#include <std.h> int RBS (int A[], int l, int n, int key); int mid; if (A[mid] == key) { printf ("mid"); } if (A[mid] < key) return RBS (A, mid+1, n, key); else if (A[mid] > key) return RBS (A, l, mid-1, key); else return -1; } int main () { int A[] = {2, 3, 4, 12, 25}; int n = 5; int result = RBS (A, 0, n-1); if (result == -1) printf ("Element is not present in array"); }	

Teacher's Signature:

Date _____

Expt. No. _____

Page. No. 14.

else

printf("Element is present at index %d", result);
return 0;

3.

Teacher's Signature : _____

Output

Sorted matrix:

11 12 22 25 34 44 70

Date 24/11/2022

Expt No. 12

Page No. 15

12. Write a program to implement bubble sort technique with optimized swapping.

```
#include <stdio.h>
void swap(int *xp, int *yp);
```

{

```
int temp = *xp;
*xp = *yp;
*yp = temp;
```

{

```
void bubbleSort(int arr[], int n)
```

{

```
int i, j;
for (i = 0; i < n - 1; i++)
    for (j = 0; j < n - i - 1; j++)
        if (arr[j] > arr[j + 1])
            swap(&arr[j], &arr[j + 1]);
```

{

```
void printArray (int arr[], int size)
```

{

```
int i;
for (i = 0; i < size; i++)
    printf("%d ", arr[i]);
printf("\n");
```

{

Teacher's Signature _____

Date _____

Expt. No. _____

Page. No. 16.

int main ()

{

int arr [] = { 64, 34, 25, 12, 22, 11, 96 };

int n = sizeof(arr) / sizeof(arr[0]);

bubbleSort (arr, n);

printf ("Sorted array :\n");

printArray (arr, n);

return 0;

}

Teacher's Signature : _____

Output

Sorted Array:
5 6 11 12 13.

Expt No	15	Date 06/11/2016
Page No	17	
Q1. Write a program to implement insertion sort technique.		
<pre>#include <cs50.h> void insertionSort(int arr[], int n) { int i, key, j; for (i = 1; i < n; i++) { key = arr[i]; j = i - 1; while (j >= 0 && arr[j] > key) { arr[j + 1] = arr[j]; j = j - 1; } arr[j + 1] = key; } } void printArray(int arr[], int n) { int i; for (i = 0; i < n; i++) { printf("%d ", arr[i]); } printf("\n"); }</pre>		

```
int main ()
```

```
{
```

```
    int arr [] = { 12, 11, 13, 5, 6 };
```

```
    int n = sizeof (arr) / sizeof (arr[0]);
```

```
    insertionSort (arr, n);
```

```
    printf (" Sorted Array :- \n ");
```

```
    printArray (arr, n);
```

```
    return 0;
```

```
}
```

Teacher's Signature : _____

Output

Sorted Array :-
11 16 22 25 49.

Date 20/11/2022
Expt No. 14. Page No. 14

Q4. Write a program to implement Selection Sort technique

```
#include <conio.h>
void swap( int *np, int *vp)
{
    int temp = *np;
    *np = *vp;
    *vp = temp;
}

void SelectionSort( int arr[], int n)
{
    int i, j, min_idx;
    for( i=0; i<n-1; i++)
    {
        min_idx = i;
        for( j=i+1; j<n; j++)
        {
            if( arr[j] < arr[min_idx])
            {
                min_idx = j;
            }
        }
        swap( &arr[min_idx], &arr[i]);
    }
}
```

Teacher's Signature _____

void printArray (int arr[], int size),

{

int i;

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

{ printf ("%d", arr[i]); }

printf ("\n");

}

int main ()

{

int arr [] = { 64, 25, 12, 22, 11 };

int n = sizeof(arr) / sizeof (arr[0]);

selectionSort (arr, n);

printf ("Sorted Array :-\n");

printArray (arr, n);

return 0;

}

Teacher's Signature : _____

Output

2 4 8 12 13 15

Date _____
Expt. No. 15. Page No. 24

Q5. Write a program to implement quick sort considering
last element as pivot.

```
#include <conio.h>
void qsort(int A[], int p, int r);
{
    if (l > r) {
        int q = partition(A, p, r);
        qsort(A, p, q-1);
        qsort(A, q+1, r);
    }
}

void swap (int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

int partition (int A[], int p, int r) {
    int i = p;
    int x = A[r];
    for (int j = p; j <= r-1; j++) {
        if (A[j] < x)
            i = i + 1;
        swap(&A[i], &A[j]);
    }
}
```

Teacher's Signature _____

Date _____

Expt. No. _____

Page. No. 22

swap (&A[i+1], &A[s]);
return (i+1);

3.

```
int main () {  
    int i;  
    int arr[] = {15, 10, 13, 9, 12, 7};  
    q_s (arr, 0, n-1);  
    for (i=0; i<n; i++) {  
        printf ("%d", arr[i]);  
    }  
    return 0;  
}
```

Teacher's Signature : _____

Output

1 3 5 6 8 10 11 13

Date 25/11/2020
Expt. No. 16 Page No. 23

Q6 Write a program to implement Queue List (including first element as front).

#include < stdio.h >

void q1(int n1, int p, int r)

{ if (p > r)

int q = permutation (n, p, r);

q1(n, p+1, r);

q1(n, p+1, r);

}

void swap (int *a, int *b)

{ int temp = *a;

*a = *b;

*b = temp;

}

int permutation (int n1, int p, int r)

{ if (p > r)

int m = RPN();

int t = p;

for (i = p + 1; i <= r; i++)

{ if (RPN() <= n)

i = i + 1;

swap (SAL[i], SAL[t]);

}

}

Teacher's Signature _____

```
swap (&A[i], &A[b]);  
return i;  
}  
int main () {  
    int i;  
    int arr [] = { 6, 10, 13, 5, 8, 3, 2, 11 };  
    qsort (arr, 0, n-1);  
    for (i=0; i<n; i++) {  
        printf ("%d", arr[i]);  
    }  
    return 0;  
}
```

Teacher's Signature : _____

Output

2 3 5 6 7 8 10 12

Date 20/1/2018
Expt. No. 19 Page No. 25

Ques. write a program to implement Merge sort.

#include <iostream>

void merge(int a[], int p, int m, int r);
int n = m - p + 1;

int m = r - m;

int left = p, right = m + 1;

for (int i = 0; i < n; i++) {
 left[i] = a[p + i];

}

for (int j = 0; j < m; j++) {
 right[j] = a[m + j + 1];

}

int i = 0, j = 0, k = p;

while (k < n) {
 if (left[i] < right[j]) {

a[k + i] = left[i];

}

else {

a[k + i] = right[j];

}

i++;

a[k + i] = left[i];

}

while (j < m) {

a[k + i] = right[j];

}

Teacher's Signature

```
void mergesort (int arr[], int l, int r) {
```

```
if (l < r) {
```

```
int m = (l+r)/2;
```

```
mergesort (arr, l, m);
```

```
mergesort (arr, m+1, r);
```

```
merge (arr, l, m, r);
```

```
}
```

```
y
```

```
int main () {
```

```
int i;
```

```
int arr[8] = {4, 2, 5, 8, 12, 3, 6, 7, 10};
```

```
mergesort (arr, 0, 7);
```

```
for (i=0; i<8; i++) {
```

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

```
}
```

```
return 0;
```

```
3.
```

Teacher's Signature : _____

Output

2 3 13 39 54

Expt No.	18	Date	23/11/2022
Page No.	3		
Ques.	Write a program to implement Shell sort.		
	#include <iostream> using namespace std; int bubbleSort(int arr[], int n) { for (int gap = n/2; gap>0; gap/=2) { for (int i = gap; i < n; i += 1) { int temp = arr[i]; int j; for (j = i - gap; j >= 0; j -= gap) { if (arr[j] > temp) { arr[j + gap] = arr[j]; } else { arr[j + gap] = temp; break; } } arr[j] = temp; } } }		
	void printArray(int arr[], int n) { for (int i = 0; i < n; i += 1) { cout << arr[i] << " "; } }		
	int main() { int arr[] = { 12, 34, 14, 2, 39 }; int n = sizeof(arr) / sizeof(arr[0]); bubbleSort(arr, n); printArray(arr, n); return 0; }		Teacher's Signature _____

Output
2 34 45 46 45 To 100 342

Date 21/11/2020
Expt. No. 14. _____
Page No. 29

Q) write A program to implement Radix Sort.

```
#include <iostream>
using namespace std;
int getMax(int arr[], int n)
{
    int max = arr[0];
    for (int i = 1; i < n; i++)
        if (arr[i] > max)
            max = arr[i];
    return max;
}

void radixSort(int arr[], int n, int digit)
{
    int output[n];
    int count[10] = {0};
    for (int i = 0; i < n; i++)
        count[(arr[i] / digit) % 10]++;
    for (int i = 1; i < n; i++)
        count[i] += count[i - 1];
    for (int i = n - 1; i >= 0; i--) {
        output[count[i] - 1] = arr[i];
        count[(arr[i] / digit) % 10]--;
    }
    for (int i = 0; i < n; i++)
        arr[i] = output[i];
}

void printArr(int arr[], int n)
{
    int max = getMax(arr, n);
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
}
```

Teacher's Signature: _____

```
void print (int arr[], int n) {  
    for (int i=0; i<n; i++)  
        cout << arr[i] << " " ;  
}
```

```
int main () {
```

```
    int arr[] = { 170, 45, 75, 90, 802, 24, 2, 66 } ;  
    int n = sizeof (arr) / sizeof (arr[0]);  
    radixsort (arr, n);  
    print (arr, n);  
    return 0;
```

```
3.
```

Output
1 2 3 3 4 2

Expt No. 20:

Date 21/11/2021

Page No. 20

To Write a program to implement Count sort.

```
#include <iostream>
using namespace std;
void countSort(int arr[], int size) {
    int output[10], count[10];
    int max = arr[0];
    for (int i=1; i<size; i++) {
        if (arr[i] > max)
            max = arr[i];
    }
    for (int i=0; i<max+1; i++)
        count[i] = 0;
    for (int i=0; i<size; i++) {
        count[arr[i]]++;
    }
    for (int i=0; i<max+1; i++)
        count[i] = count[i-1];
    for (int i=max-1; i>=0; i--) {
        output[count[i]-1] = arr[i];
        count[i] -= 1;
    }
    for (int i=0; i<size; i++)
        arr[i] = output[i];
}
```

Teacher's Signature:

void printArray (int arr[], int size) {

 for (int i=0 ; i<size ; i++) {

 cout << array [i] << " ";

 cout << endl ;

}

int main () {

 int array [] = { 4, 2, 2, 8, 3, 3, 1 } ;

 int n= size (array) / size (array [0]);

 cout << n ;

 printArray (array , n);

3.

Teacher's Signature : _____

Output

5 6 7 11 12 13

Date : 11/11/2020
Expt. No. 2.1 Page No. 2.1

Q1. Write a program to implement HeapSort.

```
#include <iostream>
using namespace std;
void heapify( int arr[], int n, int i ) {
    int largest = i;
    int l = 2 * i + 1;
    int r = 2 * i + 2;
    if (l < n && arr[l] > arr[largest])
        largest = l;
    if (r < n && arr[r] > arr[largest])
        largest = r;
    if (largest != i) {
        swap( arr[i], arr[largest] );
        heapify( arr, n, largest );
    }
}
```

```
void heapSort( int arr[], int n ) {
    for (int i = n / 2 - 1; i >= 0; i--) {
        heapify( arr, n, i );
    }
    for (int i = n - 1; i >= 0; i--) {
        swap( arr[0], arr[i] );
        heapify( arr, i, 0 );
    }
}
```

Teacher's Signature:

~~2) Write a program to implement Heap Sort.~~

```
void printArray (int arr[], int n) {
    for (int i=0; i<n; ++i) {
        cout << arr[i] << " ";
        cout << "\n";
    }
}

int main () {
    int arr[] = {12, 11, 13, 5, 6, 7};
    int n = sizeof(arr)/sizeof(arr[0]);
    heapSort (arr, n);
    printArray (arr, n);
}
```

Teacher's Signature :

Output

1. Add node at last in Link List.
2. View in Link List.
3. Delete Node at first in Link List.
4. Insert Node at First.
5. To see the length of LL.
6. To insert after position of node in LL.
7. To delete at last position of node in LL.
8. To delete at given position of node in LL.
9. Exit.

Enter the choice:

1

Enter the data made in node.

22

1. Add node at last in Link List.
2. View in Link List.
3. Delete Node at First in Link List.
4. Insert Node at First.
5. To see the length of LL.
6. To insert after position of node in LL.
7. To delete at last position of node in LL.
8. To delete at given position of node in LL.
9. Exit.

Enter the choice:

4

Enter the data made in node.

73

Date / Month / Year

Page No. 24

Expt. No. 22.

(insertion and deletion)

```

22 write a program to implement singly linked list.

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

struct node {
    int info;
    struct node * next;
};

int count = 0;
struct node * start = NULL;
struct node * currentnode();
void insert();
void insertatposition();
void insertatlast();
void deletemain();
void deleteatposition();
int getlength();
void deletemain();
struct node * createnode();
struct node * n;
n = currentnode();
n-> information = malloc(sizeof(struct node));
return n;
}

void insert()
{
    struct node * temp;
    struct node * temp1;
}

```

Teacher's Signature _____

1. Add node at last in Link List
2. view in Link List
3. Delete node at First in Link List
4. Insert node at First
5. To find the length of LL
6. To insert after position of node in LL
7. To delete at last position of node in LL
8. To delete at given position of node in LL
9. Exit

Enter the choice:

2

Linked List contains:

78

Linked List contains:

12

1. Add node at last in Link List
2. view in Link List
3. Delete node at First in Link List
4. Insert node at :
- 5.
- 6.
- 7.
- 8.
- 9.

Enter the choice:

3

Expt. No. _____	Date _____
Page No. 25	
<pre> temp = createnode(); printf ("Enter the data made in node %d",); scanf ("%d", &temp->info); temp->link = NULL; if (start == NULL) start = temp; else { t = start; while (t->link != NULL) t = t->link; t->link = temp; } } void deletemain () { // delete at first if (start == NULL) printf ("List is empty"); else { struct node *q; q = start; start = start->link; free (q); } } </pre>	
Teacher's Signature: _____	

```

void viewlist () {
    struct node *t;
    if (start == NULL) {
        printf ("no list exists\n");
    } else {
        t = start;
        while (t != NULL) {
            printf ("Linked List contains:\n");
            printf ("%d\n", t->info);
            t = t->link;
        }
    }
}

```

```

void insertfirst () {
    struct node *newnode;
    newnode = createnode();
    printf ("Enter the data in node.\n");
    scanf ("%d", &newnode->info);
    if (start == NULL) {
        start = newnode;
    } else {
        newnode->link = start;
        start = newnode;
    }
}

```

```

int getLength() {
    struct node *t;
    if (lstart == NULL) {
        printf ("empty list");
    } else {
        t = lstart;
        while (t != NULL) {
            count++;
            t = t->link;
        }
    }
    return count;
}

void insertatposition() {
    int position;
    struct node *newnode, *t;
    if (lstart == NULL) {
        printf ("No list is there");
    } else {
        printf ("please give position after you want to add new node");
        scanf ("%d", &position);
        if (position > getLength ()) {
            printf ("invalid entry");
            insertatposition(); // recursive
        }
    }
}

```

Teacher's Signature : _____

```

else {
    int i = 1;
    newnode = createnode();
    printf("give data \n");
    scanf("%d", &(newnode->info));
    newnode->link = NULL;
    if (start == NULL)
        start = newnode;
    else {
        t = start;
        while (i < position) {
            t = t->link;
            i++;
        }
        newnode->link = t->link;
        t->link = newnode;
    }
}

void deletenode() {
    struct node *t, *q;
    if (start == NULL) {
        printf("List is empty \n");
    }
}

```

Teacher's Signature : _____

else. {

t = Start;

q = Start;

while (t → Link != NULL). {

q = t;

t = t → Link;

}

if (t == Start). {

Start == NULL;

},

else. {

q → Link = NULL;

free(t);

}

,

},

void deleteatposition () . {

int position ;

struct node *t, *q;

If (Start == NULL).

printf ("No List is there.");

else. {

printf ("Please give position after you want to delete
new node.");

scanf ("%d", &position);

Teacher's Signature : _____

```

if (position > getLength ()) {
    printf ("invalid entry \n");
    deleteAtPosition ();
}

else {
    int i = 1;
    t = start;
    while (i < position - 1). {
        t = t->link;
        i++;
    }

    q = t->link;
    t->link = q->link;
    free (q);
}

int menu () {
    int ch;
    printf ("1. Add node at last in link list\n");
    printf ("2. View in link list\n");
    printf ("3. Delete node at first in link list\n");
    printf ("4. Insert node at first\n");
    printf ("5. To see the length of LL\n");
    printf ("6. To insert after position of node in
LL\n");
}

```

Teacher's Signature : _____

~~printf("%d\n",~~ 7. To delete at last position of node in LL\n");
printf("%d\n", 8. To delete at given position of node
in LL\n");

```
printf("Enter the choice\n");
scanf("%d", &ch);
return(ch);
```

1

```
void main () {
```

int Kj

while (1);

switch (menu) {

case 1:

insertnode();

break:

Case 2:

View Hist();

break:

case 3:

`steletenode();`

Boxfolk's

case 4:

`insertatfirst()`.

bryeak's

Cabe 5 :

K = getLength(l);

Teacher's Signature : _____

printf ("the length of LL is %d \n", k);
break;

case 6:

insertAfterPosition (L);
break;

case 7:

deleteAtFirst ();
break;

case 8:

deleteAtPosition ();
break;

case 9:

exit (0);
break;

default :

printf ("get lost \n");

4

3.

Output

Select the choice of operation on link 100-110.

1. Insert at begin.
2. Insert at end.
3. Insert at middle.
4. Delete from begin.
5. Delete from end.
6. Delete from middle.
7. display list.
8. exit.

Enter the choice of operation you want to do 2.
Enter the value you want to insert in node 50
 $\rightarrow 50 \rightarrow 15 \rightarrow 40$

Enter the choice of operation you want to do 3.
Enter the value you want to insert in node at last 40
 $\rightarrow 50 \rightarrow 15 \rightarrow 40 \rightarrow 45$

Enter the choice of operation you want to do 4.
After which data you want to insert 50
Enter the data you want to insert in List 25
 $\rightarrow 50 \rightarrow 25 \rightarrow 40 \rightarrow 45$

Enter the choice of operation you want to do 5.
Enter the value you want to insert in node at 100-110
 $\rightarrow 50 \rightarrow 25 \rightarrow 40 \rightarrow 45 \rightarrow 55$

Date 24/01/2016

Page No. 47

Expt No. 2.1

2.1 write a program to implement insertion and deletion using doubly linked list.

```
#include <stdio.h>
#include <conio.h>
struct node {
    struct node * previous;
    int data;
    struct node * next;
};

struct node * head;
struct node * last;
void insertion_beginning();
void insertion_end();
void insertion_specified();
void deletion_beginning();
void deletion_specified();
void display();
void insertion_end() {
    struct node * newnode;
    newnode = (struct node *) malloc(1 * sizeof(struct node));
    newnode->data = value;
    newnode->previous = NULL;
    newnode->next = NULL;
    if (head == NULL) {
        head = last = newnode;
    } else {
        last->next = newnode;
        newnode->previous = last;
        last = newnode;
    }
}
```

Teacher's Signature:

Enter the choice of operation you want to do 4.

Node deleted.

$\rightarrow 25 \rightarrow 48 \rightarrow 47$

Enter the choice of operation you want to do 5

Data deleted from list to 47.

$\rightarrow 25 \rightarrow 48$

Enter the choice of operation you want to do 7.

$\rightarrow 25 \rightarrow 48$

insertion

Date _____
Expt. No. _____ Page No. 44

else if

head->previous = newnode;

newnode->next = head;

head = newnode;

3

void insertion(int value) {

struct node *newnode;

newnode = (struct node *) malloc (sizeof (struct node));

newnode->data = value;

newnode->previous = NULL;

newnode->next = NULL;

If (head == NULL) {

head = last = newnode; }

4

else if

last->next = newnode;

newnode->previous = last;

last = newnode;

5

void insertion specified (int value, int loc) {

struct node *temp; *temp, *temp1;

var = (struct node *) malloc (sizeof (struct node));

var->data = value;

Testers signature

```

if (head == NULL) {
    head = var;
    head->previous = NULL;
    head->next = NULL;
}

else {
    temp = head;
    while (temp != NULL && temp->data != loc) {
        temp = temp->next;
    }

    if (temp == NULL) {
        printf("In %d is not present in List.", loc);
    }

    else {
        temp1 = temp->next;
        temp->next = var;
        var->previous = temp;
        var->next = temp1;
        temp1->previous = var;
    }
}

last = head;
while (last->next != NULL) {
    last = last->next;
}

```

Teacher's Signature : _____

void deletion_beginning() {

struct node *ptr;

if (head == NULL) {

printf("In Underflow");

}

else if (head->next == NULL) {

head = NULL;

free(head);

printf("In node deleted\n");

}

else {

ptr = head;

head = head->next;

head->previous = NULL;

free(ptr);

printf("In node deleted\n");

}

}

void deletion_last() {

struct node *temp;

temp = last;

if (temp->previous == NULL) {

free(temp);

head = NULL;

last = NULL;

}

Teacher's Signature : _____

```

printf ("Data deleted from list is %d\n", last->data);
last = temp->previous;
last->next = NULL;
free (temp);
}

```

void deletion (specified (int value)) {

```

struct node *temp, *var, *t, *temp1;
temp = head;

```

```

while (temp != NULL) {
    if (temp->data == value) {
        if (temp->previous == NULL) {
            free (temp);
            head = NULL;
            last = NULL;
        }
    }

```

```

    else {

```

```

        var->next = temp1;
    }

```

```

    temp1->previous = var;
    free (temp1);
}

```

```

}

```

```

else {

```

```

    var = temp;
}

```

```

    temp = temp->next;
}

```

```

temp1 = temp->next;
}

```

```

}

```

Teacher's Signature : _____

```

printf ("data deleted from %12s is %d", value);
}

void display() {
    struct node *temp;
    temp = head;
    if (temp == NULL) {
        printf ("List is Empty");
    }
    while (temp != NULL) {
        printf ("%12s", temp->data);
        temp = temp->next;
    }
}

int main() {
    int value, i, loc;
    head = NULL;
    printf ("Select the choice of operation on link list ");
    printf ("\n 1. insert at begining");
    printf ("\n 2. insert at end");
    printf ("\n 3. insert at middle");
    printf ("\n 4. delete from begining");
    printf ("\n 5. delete from end");
    printf ("\n 6. delete from middle");
    printf ("\n 7. display list");
    printf ("\n 8. exit");
}

```

Teacher's Signature : _____

```
while (1) {
```

```
    printf ("\\n\\nEnter the choice of operation you want\n"
           "to do \\n");
```

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

```
    switch (i) {
```

```
        case 1: {
```

```
            printf ("\\nEnter the value you want to insert in\n"
                    "node \\n");
```

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

```
            insertion_beginning (value);
```

```
            display ();
```

```
            break;
```

```
        }
```

```
        case 2: {
```

```
            printf ("\\nEnter the value you want to insert in\n"
                    "node at (int \\n);
```

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

```
            insertion_last (value);
```

```
            display ();
```

```
            break;
```

```
        }
```

```
- Case 3: {
```

```
    printf ("\\nEnter which data you want to insert data \\n");
```

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

```
    printf ("\\nEnter the data you want to insert in \\n");
```

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

Teacher's Signature: _____

-insertion - specified (value, n.c.) ;
 display ();
 break;

4.

case 4 : {

deletion - beginning ();
 display ();
 break;

4

case 5 : {

deletion - last ();
 display ();
 break;

4

case 6 : {

printf ("enter the value you want to delete ");
 scanf ("%d", &value);
 deletion - specified (value);
 display ();
 break;

4

case 7 : {

display ();
 break;

4

Date _____

Expt. No. _____

Page. No. 51

case 8 : {,

 exit(0);

 break;

}

}

g.

 printf("In Instd"); last → date();

 display();

g.

Teacher's Signature : _____

Ques 10
Choose one option from the following List:-

1. Insert in beginning
2. Insert at last
3. Delete from Beginning
4. Delete from last
5. Search for an element
6. Show
7. Exit

Enter your choice:

2.

Enter the node data? 51
node inserted.

Enter your choice:

1.

Enter the node data? 78
node inserted.

Enter your choice:

t

Enter Data? 11
node inserted.

Date 24/12/2020

Expt No. 24

Page No. 51

```
#include <stdio.h>
#include <stdlib.h>
struct node {
    int data;
    struct node *next;
};

void insert();
void display();
void search();
void begin();
void last();
void insert();
void display();
void search();
void begin();
void last();
int main() {
    struct node *ptr = malloc(sizeof(struct node));
    if (ptr == NULL) {
        printf("Memory allocation error");
    }
    else {
        printf("Enter the node data:");
        scanf("%d", &ptr->data);
        ptr->next = NULL;
        insert(ptr);
        display();
    }
}
```

Teacher's Signature

Enter your choice:
3

Enter Data : 15
node inserted.

Enter your choice:
4

printing values -->
44
58
11
65

Enter your choice:
3

node deleted.

Enter your choice:
4

node deleted.

Enter your choice:
6

printing values -->
58
11

Date _____
Expt. No. _____ Page No. 53

```
preAllocate a item;
if (head == NULL) {
    head = ptx;
    ptx->next = head;
}
else {
    temp = head;
    while (temp->next != head)
        temp = temp->next;
    temp->next = ptx;
    head = ptx;
}
printf("The node insertion is %d\n");
}

void lastDelete() {
    struct node *ptr, *temp;
    int item;
    ptr = lastStruct->node + 1 - value(lastStruct->node);
    if (ptr == NULL) {
        printf("The insertion is not possible\n");
    }
    else {
        printf("The insertion is %d\n");
        item = lastStruct->node + 1 - value(ptr);
        temp = lastStruct->node + 1 - value(item);
    }
}
```

Teacher's Signature: _____

Enter your choice :

5

Enter Item which you want to search ?

75

Item not found.

Programmer : [Signature]

Date : [Signature]

Dept. No.	Date
	Page No. 54
<pre>preodeate = item; if (head == NULL) { head = preodeate; preodeate->next = head; } else { temp = head; while (temp->next != head) { temp = temp->next; } temp->next = preodeate; preodeate->next = head; } printf("%n node inserted n"); } void display() { struct node *ptr; if (head == NULL) { printf("List is empty"); } else if (head->next == head) { head = NULL; free(head); printf("List is empty"); } }</pre>	
Teacher's Signature : [Signature]	

else {

ptr = head;

while (ptr->next != head) :

ptr = ptr->next;

ptr->next = head->next;

free (head);

head = ptr->next;

printf ("A node deleted\n");

}

,

void last_delete () {

struct node *ptr, *preptr;

if (head == NULL) {

printf ("Underflow");

,

else if (head->next == head) {

head = NULL;

free (head);

printf ("A node deleted\n");

,

else {

ptr = head;

while (ptr->next != head) {

preptr = ptr;

ptr = ptr->next;

,

Teacher's Signature : _____

```

prenode = ptr->next;
free(ptr);
printf("A node deleted.\n");
}

```

4.

```
void search();
```

```

struct node *ptr;
int item, i=0, flag=1;
ptr = head;
if (ptr == NULL) {

```

```

    printf("An Empty List.\n");
}

```

```
else {

```

```

    printf("Enter item which you want to search ?\n");
    scanf("%d", &item);

```

```

    if (head->data == item) {

```

```

        printf("Item found at location %d", i+1);

```

```

        flag = 0;
}

```

```
else {

```

```

    while (ptr->next != head) {

```

```

        if (ptr->data == item) {

```

```

            printf("Item found at location %d", i+1);

```

```

            flag = 0;

```

```

            break;
}

```

Teacher's Signature : _____

else {

 flag = 1;

}

 i++;

 ptr = ptr -> next;

}

}

if (flag != 0) {

 printf("Item not found \n");

}

}

}

void display() {

 struct node *ptr;

 ptr = head;

 if (head == NULL) {

 printf("nothing to print");

}

 else {

 printf("\n printing value ... \n");

 while (ptr->next != head) {

 printf("%d\n", ptr->data);

 ptr = ptr->next;

}

 printf("%d\n", ptr->data);

}

}

Teacher's Signature : _____

```

void main() {
    int choice = 0;
    printf("1. choose one option from the following list... \n");
    printf("1. Insert in begining \n 2. Insert at last \n");
    printf("3. Delete from beginning \n 4. Delete from last \n");
    printf("5. Search for an element \n 6. Show list. Exit \n");
    while (choice != 7) {
        printf("Enter your choice ? ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                beginInsert();
                break;
            case 2:
                lastInsert();
                break;
            case 3:
                beginDelete();
                break;
            case 4:
                lastDelete();
                break;
            case 5:
                search();
                break;
        }
    }
}

```

Teacher's Signature: _____

Date _____

Expt. No. _____

Page. No. 59.

case 6:

display();

break;

case 7:

init();

break;

default:

printf("Please enter valid choice.. ");

y

y

y

Teacher's Signature : _____

```

--- Enter choice ---
1. push
2. pop
3. peek
4. display
5
Enter the data:
25
--- Enter choice ---
1. push
2. pop
3. peek
4. display
1
Enter the data:
25
--- Enter choice ---
1. push
2. pop
3. peek
4. display
1
Enter the data:
25
--- Enter choice ---
1. push
2. pop
3. peek
4. display

```

Expt No. 25

Date 24/12/2020

Page No. 60

Q5. write a program to implement stack using array.

```

#include < stdio.h >
#include < stdlib.h >
#define n 5
int stack[n];
int top = -1;
void push()
{
    if(top == n-1)
        printf("Stack overflow\n");
    else
        top++;
    stack[top] = n;
}
void pop()
{
    if(stack == -1)
        printf("Underflow\n");
    else
        printf("%d\n", stack[top]);
    top--;
}

```

Teacher's Signature

3
25
--- enter choice ---
1. push
2. pop
3. peek
4. display
4
15

29
--- enter choice ---
1. push
2. pop
3. peek
4. display
2
25
--- Enter choice ---
1. push
2. pop
3. peek
4. display
4

29

Date _____
Expt. No. _____ Page No. 51

1.
printf("Enter item\n");
2. void peek() {
 if (top == -1) {
 printf("Stack is empty.\n");
 } else {
 printf("%d\n", stack[top]);
 }
}
3.
void display() {
 int i;
 for (i = top; i >= 0; i--) {
 printf("%d\n", stack[i]);
 }
}
4.
void main() {
 int ch;
 do {
 printf(" --- Enter choice ---\n");
 printf("1. to push\n");
 printf("2. to pop\n");
 printf("3. to peek\n");
 printf("4. to display\n");
 printf("\nEnter choice\n");
 scanf("%d", &ch);
 } while (ch != 5);
}

Teacher's Signature: _____

```
scanf("%d", &ch);
switch(ch) {
    case 1: push();
    break;
    case 2: pop();
    break;
    case 3: peek();
    break;
    case 4: display();
    break;
    default:
        printf("invalid choice\n");
}
while(ch!=0);
```

Teacher's Signature: _____

Output

7
9
8
5

the pop element is : 7.
top element is : 9.
9
8
5.

Date 24/10/2020
Expt. No. 26 Page No. 51

26. Write a program for implement stack using linked list.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
struct node {
    int data;
    struct node *next;
};
struct node *top = NULL;
void push() {
    struct node *newnode;
    newnode = (struct node *)malloc(sizeof(struct node));
    newnode->data = x;
    newnode->next = top;
    top = newnode;
}
void pop() {
    struct node *temp;
    temp = top;
    if (top == NULL) {
        printf("Underflow\n");
    } else {
        printf("the pop element is : %d\n", top->data);
        top = top->next;
    }
}
```

Teacher's Signature _____

```

        free(temp);
    }

3.
void peek() {
    if (top == NULL) {
        printf("Underflow\n");
    } else {
        printf("Top element is : %d\n", top->data);
    }
}

4.
void display() {
    struct node *temp;
    temp = top;
    if (top == NULL) {
        printf("Underflow\n");
    } else {
        while (temp != NULL) {
            printf("%d\n", temp->data);
            temp = temp->next;
        }
    }
}

```

Teacher's Signature : _____

Date _____

Expt. No. _____

Page. No. 65

void main () {

push (5);

push (8);

push (9);

push (7);

display ();

pop ();

peek ();

display ();

}

Teacher's Signature : _____

Output:

```

queue is 2
queue is 3
queue is 8
queue is -1
front value is 2
dequeue element is 2
front value is 3
queue is 3
queue is 8
queue is -1

```

Q3. write a program to implement queue using array.

```

#include <stdio.h>
#include <conio.h>
#define N 5
int queue[N];
int front = -1;
int rear = -1;
void enqueue(int n) {
    if (rear == N - 1) {
        printf("Queue Overflow");
    }
    else if (front == -1 && rear == -1) {
        front = rear = 0;
        queue[rear] = n;
    }
    else {
        rear++;
        queue[rear] = n;
    }
}
void dequeue() {
    if (front == -1 && rear == -1) {
        printf("Underflow");
    }
    else if (front == rear) {
        front = rear = -1;
    }
}

```

Author's Signature

```

else {
    printf("dequeue element is %d\n", queue[front]);
    front++;
}

void display() {
    int i;
    if (front == -1 && rear == -1) {
        printf("Underflow\n");
    }
    else {
        for (i = front; i < rear + 1; i++) {
            printf("queue[%d] is %d\n", i, queue[i]);
        }
    }
}

void peek() {
    if (front == -1 && rear == -1) {
        printf("Underflow\n");
    }
    else {
        printf("peek-value is %d\n", queue[front]);
    }
}

```

Teacher's Signature : _____

```
void main() {  
    enqueue(12);  
    enqueue(3);  
    enqueue(18);  
    enqueue(11);  
    display();  
    peek();  
    dequeue();  
    peek();  
    display();  
}
```

Teacher's Signature : _____

Output

```
queue is 5  
queue is 8  
queue is 9  
queue is 10  
the peek value is 5  
the element dequeued is  
the peek value is 8  
queue is 8  
queue is 9  
queue is 10
```

Date 14/11/2012
Expt No. 3.3 Page No. 69

```
#include <iostream.h>  
#include <iomanip.h>  
#include <conio.h>  
struct node {  
    int data;  
    struct node *next;  
};  
struct node *front = NULL;  
struct node *rear = NULL;  
void Enqueue(int n);  
struct node *remove();  
remove = (struct node *) malloc(sizeof(struct node));  
remove->data = n;  
remove->next = NULL;  
if (front == NULL && rear == NULL) {  
    front = rear = remove;  
}  
else {  
    rear->next = remove;  
    rear = remove;  
}  
void display();  
struct node *temp;
```

Teacher's Signature:

```

if (front == NULL && rear == NULL) {
    printf ("queue is empty \n");
}
else {
    temp = front;
    while (temp != NULL) {
        printf ("queue %d \n", temp->data);
        temp = temp->next;
    }
}

```

```
void dequeue() {
```

```
struct node * temp;
```

```
temp = front;
```

```
if (front == NULL && rear == NULL) {
```

```
    printf ("queue is empty \n");
}

```

```
else {
```

```
    printf ("the element dequeued %d \n", front->data);
```

```
    front = front->next;
```

```
    free (temp);
}

```

```
}
```

```
void peek() {
```

```
if (front == NULL && rear == NULL) {
```

```
    printf ("queue is empty \n");
}

```

Teacher's Signature : _____

else {

printf("the peek ~~is~~ value is %d \n", front->data);
3.

};

void main() {

enqueue(5);

enqueue(18);

enqueue(19);

enqueue(10);

display();

peek();

dequeue();

peek();

display();

3.

Teacher's Signature : _____

Output

Queue is :

2

-1

5

6

4.

The dequeue element is 2.

The peek value is -1

The dequeue element is -1.

Queue is :

5

6

4.

.

Date 24/11/2016

Expt No. 39

Page No. 31

Q9. Write a program to implement circular queue using array.

```
#include <stdio.h>
#include <stdlib.h>
#define N 5
int queue[N];
int front=-1;
int rear=-1;
void enqueue(int x) {
    if(front == -1 && rear == -1) {
        front = rear = 0;
        queue[rear] = x;
    }
    else if((rear+1)%N == front) {
        printf("Overflow\n");
    }
    else {
        rear = (rear+1)%N;
        queue[rear] = x;
    }
}
void dequeue() {
    if(front == -1 && rear == -1) {
        printf("Underflow\n");
    }
    else if(front == rear) {
        front = rear = -1;
    }
}
```

Teacher's Signature

else {

```
printf("The dequeued element is %.d\n", queue[front]);
front = (front + 1) % N;
```

}

void display () {

```
int i = front;
```

```
if (front == -1 && rear == -1) {
```

```
printf("Underflow\n");
```

}

else {

```
printf("Queue is : \n");
```

```
while (i != rear) {
```

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

```
i = (i + 1) % N;
```

}

```
printf("\n%.d\n", queue[rear]);
```

}

,

void peek() {

```
if (front == -1 && rear == -1) {
```

```
printf("Underflow\n");
```

}

else {

```
printf("The peek value is %.d\n", queue[front]);
```

}

.

Teacher's Signature : _____

Date _____

Expt. No. _____

Page. No. 74.

void main() {

enqueue(2);

enqueue(-1);

enqueue(5);

enqueue(6);

enqueue(7);

display();

dequeue();

peek();

dequeue();

display();

}

Teacher's Signature : _____

Output:

Queue is :

3

-1

5

The element dequeued is :

The peek value is -1.

Queue is :

-1

5

Expt. No. 30.	Date 04/04/2020
Page No. 35.	
30. Write a C program to implement circular queue using structures.	
#include <stdio.h>	
#include <stdlib.h>	
#include <conio.h>	
struct node {	
int data;	
struct node *next;	
};	
struct queue {	
struct node *front = NULL;	
struct node *rear = NULL;	
};	
void Enqueue(queue *q, int n)	
{	
struct node *newnode;	
newnode = (struct node *) malloc(sizeof(struct node));	
newnode->data = n;	
newnode->next = NULL;	
if (front == NULL) {	
front = rear = newnode;	
}	
else {	
rear->next = newnode;	
rear = newnode;	
}	
}	
void Dequeue(queue *q)	
{	
struct node *temp;	
temp = front;	
front = front->next;	
free(temp);	
}	
void Display(queue *q)	
{	
struct node *temp;	
temp = front;	
while (temp != NULL) {	
printf("%d ", temp->data);	
temp = temp->next;	
}	
}	
}	

Student's Signature

```

void display () {
    struct node * temp;
    temp = front;
    if (front == NULL && rear == NULL) {
        printf ("queue is empty \n");
    }
    else {
        printf ("queue is : \n");
        while (temp->next != front) {
            printf ("%d \n", temp->data);
            temp = temp->next;
        }
        printf ("%d \n", temp->data);
    }
}

```

```

void dequeue () {
    struct node * temp;
    temp = front;
    if (front == NULL && rear == NULL) {
        printf ("queue is empty \n");
    }
    else if (front == rear) {
        front = rear = NULL;
        free (temp);
    }
}

```

else {

```
printf("the element dequeued is %d\n", front->data);
front = front->next;
rear->next = front;
free(temp);
```

}

},

void peek() {

```
if (front == NULL && rear == NULL) {
    printf("queue is empty\n");
}
```

else {

```
printf("the peek value is %d\n", front->data);
```

}

}

void main() {

```
enqueue(2);
```

```
enqueue(4);
```

```
enqueue(5);
```

```
display();
```

```
dequeue();
```

```
peek();
```

```
display();
```

}

Teacher's Signature : _____

Output:

- 1- Enqueue element into Queue.
 - 2- Dequeue element from Queue.
 - 3- Display from queue.
 - 4- Exit.
- Enter choice 1.
Enter data into Queue 15.
- Enter choice 2.
Enter data into Queue 45.
- Enter choice 3.
Enter data into Queue 23.
- Enter choice 4.
15 45 23.
- Enter choice 2.
15 45 23.

Dept No.	81	Date 21/01/2024
Page No.	18	
<pre>11 write = <program>{C...Implement Queue using class {public <function> {display <function> void push1(int); void push2(int); int pop1(); int pop2(); void enqueue(); void dequeue(); void display(); void create(); int m1(m1 >> m2); int top1 = -1, top2 = -1; int count = 0; void main(); int ans; push1(); // 1 - Enqueue element into Queue; push1(); // 2 - Dequeue element from Queue; push1(); // 3 - Display from Queue; push1(); // 4 - Exit(); return 0; }main(); //Input: 11 in Enter choice //Output: 15 45 23 //Author: shubham //Date: 21/01/2024</pre>		
		Teacher's Signature

case 1:

```
    enqueue();
    break;
```

case 2:

```
    dequeue();
    break;
```

case 3:

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

case 4:

```
    exit(0);
```

default:

```
    printf("Wrong choice.");
```

}

{

void create() {

```
    top1 = top2 = -1;
```

}

void push1(int data) {

```
    A[++top1] = data;
```

}

int pop1() {

```
    return A[top1--];
```

}

Teacher's Signature : _____

```
void push2 (int data) {
```

```
    A[2][++top2] = data;
```

```
}
```

```
int pop2 () {
```

```
    return (A[2][top2 - 1]);
```

```
}
```

```
void enqueue () {
```

```
    int data, i;
```

```
    printf ("Enter data into queue ");
```

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

```
    push1 (data);
```

```
    count++;
```

```
}
```

```
void dequeue () {
```

```
    int i;
```

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

```
        push2 (pop1());
```

```
}
```

```
    pop2();
```

```
    count--;
```

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

```
        push1 (pop2());
```

```
}
```

```
}
```

```
void display () {
```

```
    int i;
```

```
    for (i = 0; i <= top1; i++) {
```

```
        printf ("%d", s1[i]);
```

```
}
```

Teacher's Signature : _____

Output

The queue is 7 5 3 -1 0
the rear of queue is 0
the front of queue is 7.
the dequeue from front is 7.
The dequeue from rear is 0
the dequeue from front is 3
the queue is 1 -2 -1

Expt No.	31	Date	04/03/2020
Page No.	32		
Q3	WRITE a program to implement a queue (linked list). #include <stdio.h> #define n 5 int deque; int f, r; void enqueue(front, rear); if (f == n) { if (r == n - 1) { printf ("Queue is full\n"); } else if (f == -1 && r == -1) { f = 0; dequeue(f); } else { int t; t = r + 1; for (i = r; i = t - 1; i--) { dequeue(i); } dequeue(f); } } else enqueue(front, rear); if (f == n) { if (r == n - 1) { printf ("Queue is full\n"); } else { int t; t = r + 1; for (i = r; i = t - 1; i--) { dequeue(i); } dequeue(f); } } else enqueue(front, rear); if (f == n) { if (r == n - 1) { printf ("Queue is full\n"); } else { int t; t = r + 1; for (i = r; i = t - 1; i--) { dequeue(i); } dequeue(f); } }		
		Teacher's Signature	

```
else if (f == -1 && r == -1) {
```

```
    f = r = 0;
```

```
    deque[r] = n;
```

```
}
```

```
else if (r == N - 1) {
```

```
    r = 0;
```

```
    deque[r] = n;
```

```
}
```

```
else {
```

```
    r++;

```

```
    deque[r] = n;
```

```
}
```

```
}
```

```
void display() {
```

```
    int i = f;
```

```
    printf("the queue is :- ");
```

```
    while (i != r) {
```

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

```
        i = (i + 1) % N;
```

```
}
```

```
    printf("%d\n", deque[r]);
```

```
}
```

```
void getfront() {
```

```
    if (f == -1 && r == -1) {
```

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

```
}
```

Teacher's Signature : _____

else {

 printf("the front of queue is %d\n", deque[f]);

}

}

void get rear() {

 if (f == -1 && r == -1) {

 printf("queue is empty\n");

}

else {

 printf("the rear. of queue is %d\n", deque[r]);

}

}

void dequuefromfront() {

 if (f == -1 && r == -1) {

 printf("queue is empty\n");

}

else if (f == r) {

 f = r = -1;

}

else if (f == N - 1) {

 printf("the dequue from front is %d\n", deque[f]);

 f = 0;

}

else {

 printf("the dequue from front is %d\n", deque[f]);

 f++;

}

Teacher's Signature : _____

```
Void dequeuerear() {
```

```
if (f == -1 && r == -1) {
```

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

```
else if (f == r) {
```

```
    f = r = -1;
```

```
else {
```

```
    printf ("The dequeue from rear is %d\n", deque[r]);
```

```
    r = N - 1;
```

```
}
```

```
void main() {
```

```
enqueuefront(2);
```

```
enqueuefront(5);
```

```
enqueuerear(-1);
```

```
enqueuerear(0);
```

```
enqueuefront(17);
```

```
display();
```

```
getrear();
```

```
getfront();
```

```
dequeuerear();
```

```
dequeuerear();
```

```
display();
```

```
}
```

Teacher's Signature: _____

Output

1- Insert an element into Queue
2- Delete an element from Queue
3- Display Queue Elements
4- Exit.
Enter your choice : 2
Enter value to be inserted : 25

Enter your choice : 2
Enter value to be inserted : 25

Enter your choice : 2
Enter value to be inserted : 25

Enter your choice : 2
Enter value to be inserted : 25

Enter your choice : 2
Enter value to be inserted : 25

Enter your choice : 2
Enter value to be inserted : 25

Enter your choice : 2
Enter value to be deleted : 25

3 not found in Queue to delete.

Date : 24/12/2020
Expt No. : 21 Page No. : 8A

```
[1] write a program to implement a queue using Queue.h  
#include <iostream.h>  
#include <Queue.h>  
#define Max 5  
void insert(int priority);  
void delete(int priority);  
void display();  
void creat();  
void insert();  
int prique[5];  
int front, rear, s;  
creat();  
int m;  
printf("1- Enter an element from Queue");  
printf("2- Delete an element from Queue");  
printf("3- Display Queue Elements");  
printf("4- Exit Queue");  
creat();  
while(1){  
    printf("Enter your choice : ");  
    scanf("%d", &m);  
    switch(m){  
        case 1:  
            printf("Enter value to be inserted : ");  
            scanf("%d", &s);  
            insert(s);  
            break;  
        case 2:  
            printf("not found in Queue to delete");  
            break;  
        default:  
            cout<<"Wrong Input";  
            break;  
    }  
}
```

Teacher's Signature:

Index agum. chalit: 2.

Krediti vplati se izdaje. č. 35

Environ. youth choice 1.3
85 14.9 3.5

Expt. No _____ Page No. _____ Date _____

case 3.2
 prints 1st maximum value to delete : 1st
 Among 1st and 2nd + 4th,
 delete by priority 1st
 because,
 case 3.
 display - 3 4 5 6 7 8;
 because,
 case 4.1
 cout < 1st;
 display 2;
 prints 1st in choice (in inserted, both a correct
 choice 1st);
 1
 3
 1st function to create & supply priority queue 4.
 void createl() {
 front = rear = -1;
 }
 3
 1st function to insert value into priority queue 4.
 void insert(int priority 1st data) {
 if (rear == MAX - 1) {
 cout << "queue overflow no more elements can be
 inserted." >>
 return;
 }
 3
 Teachers Signature _____

```
else if (!front == -1) {if (rear == -1), {
```

```
    front++;
    rear++;
    pri-que[rear] = data;
    return;
```

```
}
```

```
else :
```

```
    check(data);
```

```
    rear++;
}
```

* Function to check priority and place element +1.

```
void check(int data){
```

```
int i, j;
```

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

```
    if (data >= pri-que[i]) {
```

```
        for (j = rear + 1; j > i; j--) {
```

```
            pri-que[j] = pri-que[j - 1];
```

```
}
```

```
        pri-que[i] = data;
```

```
        return;
```

```
}
```

```
}
```

```
pri-que[i] = data;
```

```
g
```

~~void delete_by_index~~

```
/* Function to delete an element from queue + 1 */
void delete_by_priority(int data) {
    int i;
    if (lfront == -1 && lrear == -1) {
        printf ("In queue. is empty no elements to delete ");
        return ;
    }
    for (i = 0; i <= rear; i++) {
        if (data == prique[i]) {
            for (j = i + 1; j < rear; j++) {
                prique[i] = prique[j];
            }
            prique[i] = -99;
            rear--;
            if (rear == -1) {
                front = -1;
            }
            return ;
        }
    }
    printf ("I did not found in queue to delete ", data);
}
```

1* Function - to display Queue elements *%.

void display ('queue' 1) {

if (front == -1) & (rear == -1) {

printf ("in queue is empty ");

return 0;

}

for (l; front <= rear; front++) {

printf ("%d", que[front]);

}

front = 0;

3.

Teacher's Signature : _____

Output

Preorder traversal of binary tree is -

1 2 4 5 3

Inorder traversal of binary tree is -

4 2 5 1 3

Postorder traversal of binary tree is

4 5 2 3 1

Date 24/05/2022

Expt No. 34

Page No. 10

Q4. Write a program to find inorder, preorder and postorder of a binary tree.

```
#include < stdio.h>
#include < stdlib.h>
struct node {
    int data;
    struct node * left;
    struct node * right;
};

struct node * newnode (int data) {
    struct node * node;
    node = (struct node*) malloc (sizeof (struct node));
    node->data = data;
    node->left = NULL;
    node->right = NULL;
    return (node);
}

void printpostorder (struct node * node) {
    if (node == NULL) {
        return;
    }
    printpostorder (node->left);
    printpostorder (node->right);
    printf (" %d ", node->data);
}
```

Teacher's Signature

```
Void printInorder ( struct node * node ) {
```

```
    if ( node == NULL ) {  
        return ;  
    }
```

```
    printInorder ( node -> left );  
    printf ("% .d ", node -> data );  
    printInorder ( node -> right );  
}
```

```
Void printPreorder ( struct node * node ) {
```

```
    if ( node == NULL ) {  
        return ;  
    }
```

```
    printf ("% .d ", node -> data );  
    printPreorder ( node -> left );  
    printPreorder ( node -> right );
```

```
int main () {
```

```
    struct node * root = newnode ( 1 );
```

```
    root -> left = newnode ( 2 );
```

```
    root -> right = newnode ( 3 );
```

```
    root -> left -> left = newnode ( 4 );
```

```
    root -> left -> right = newnode ( 5 );
```

```
    printf (" \n Preorder traversal of binary tree is \n " );
```

```
    printPreorder ( root );
```

```
    printf (" \n Inorder traversal of binary tree is \n " );
```

```
    printInorder ( root );
```

Teacher's Signature : _____

Date _____

Expt. No. _____

Page. No. 92.

printf("In Postorder traversal of binary tree is %n");
printpostorder(&root);
getchar();
return 0;

3

Teacher's Signature : _____