Page No.: _____

Date ___/___/_____

S.No	Program Names	Pages	Remar
1.	//WAP to show compile time Initialization	1 to 2	
2.	//WAP to show run time initialization	3 to 4	
3.	//WAP to reverse an Array	5 to 6	
4.	//WAP to input and display the array by using function.	7 to 8	
5.	//WAP to show max and min element with the help of functions.	9 to 12	
6.	//WAP to find add and even element with the help of functions.	13 to 16	
7.	//WAP to add 10 in all the elements of an array with the help of functions.	17 to 20	
8.	//WAP to show linear search by using function	21 to 24	
9.	//WAP to show binary search by using function.	25 to 28	
10.	//WAP to show bubblesort by use function.	29 to 32	
11.	//WAP for selection sort	33 to 36	
12	//WAP to show insertion sort	37 to 40	
13	//WAP for addition of two matrices	41 to 44	
14.	//WAP to find transpose of a matrix	45 to 48	
15	//WAP to multiply two matrices	49 to 52	
16.	//WAP of sparse array	53 to 56	
17.	//WAP in C to show vector representation of lower triangular sparse matrix	57 to 60	
18.	//WAP to show vector representation of upper triangular matrix	61 to 64	
19.	//WAP to show vector representation of diagonal sparse matrix	65 to 68	
20.	//WAP to show the vector representation of tridiagonal sparse matrix.	69 to 72	
21.	//WAP to merge two sorted array	73 to 78	

Date ___/___/_____ Page No.: _____

S.No	Program Names	Pages	Re
22.	//WAP to reverse a string	79 to 82	
23.	//WAP to show the use and representation of stack	83 to 86	
24.	//WAP to show peek function of stack	87 to 90	
25	//WAP to check validity of brackets in the expression.	91 to 94	
26.	//WAP to find the factorial with the help of recursion	95 to 96	
27.	//WAP to find the fibonacci series in Ds	97 to 98	
28.	//WAP to Perform the queue & Deque operation	99 to 102	
29.	//WAP to Insert the last node in linked list.		

Chitra

"No religion has mandated killing others as a requirement for its sustenance or promotion." –Dr.A.P.J.Abdul Kalam

Date / /

S

Output :-

The Value of arr[0] is : 10

The Value of arr[1] is : 38

The value of arr[2] is : 23

The value of arr[3] is : 56

The value of arr[4] is : 76

Date 31.08.24

Program - 1

```
// WAP to show compile time Initialization.

# Include <stdio.h>
# Include <conio.h>
  void main()
{
  int arr[10] = {10, 38, 23, 56, 76};
  printf("\n The value of arr[0] is: %d", arr[0]);
  printf("\n The value of arr[1] is: %d", arr[1]);
  printf("\n The value of arr[2] is: %d", arr[2]);
  printf("\n The value of arr[3] is: %d", arr[3]);
  printf("\n The value of arr[4] is: %d", arr[4]);

  getch();
}
```

Output :

Enter the size of an array 5

Enter the elements : 21
22
34
35
45

Elements Given by User:
21
22
34
35
45

Program = 2

```
// WAP to show run time initialization.
# include < stdio.h >
# include < conio.h >
void main()
{
  int arr [7], n, i;
  printf("\n Enter the size of an array");
  scanf("%d", &n);
  printf("\n Enter the elements:");
  for (i = 0; i<n ; i++)
{
  scanf( "\n %d ", &arr[i]);
}
  printf (" \n Elements Given by User:");
  for (i = 0; i<n ; i++)
{
  printf("\n %d ", arr[i] );
}
}
```

Date __/__/____

Output :-

Input no. of elements : 5

Array elements are :

2
6
8
10
12

Array after reversing :

12
10
8
6
2

Date __/__/____

Program - 3

```
//WAP to reverse an array.

# include <stdio.h>
# include < conio.h>
  void main()
{
   int arr[10], i, n ;
   printf("Input no. of elements:" );
   scanf(" %d ", &n);
   printf("Array elements are : \n");
   for (i= 0; i<n ; i++)
{
    scanf(" %d", &arr[i]);
}
   printf("Array after reversing : \n");
   for (i= n-1; i>=0; i--)
{
    printf("\n %d", a[i]);
}
}
```

Date

Output :-

```
enter no. of elements : 5
Enter elements : 45
98
65
34
21

Array elements enter by you :
45
98
65
34
21
```

Program - 4

```
//WAP to input and display the array by using
  function.
# include <stdio.h>
# include <conio.h>
 void input (int m[], int n);
 void display (int m[], int n);

 void input ( int m[], int n )
{
   int i;
   for (i=0; i<n; i++)
{
    scanf("%d", &m[i]); }
}
   void display (int m[], int n)
{
   int i;
   for (i=0 ; i<n ; i++)
{
    Printf("\n %d ", [i]); }
}
   int main() {
   int m[10], n;
   printf(" enter no. of elements:");
   scanf("%d ", &n);
   input ( m, n);
   printf(" Array elements enter by you:");
    display (m,n);
      getch ();
}
```

Program = 5

//WAP to show max and min element with the help of functions.

```
# include <stdio.h>
# include <conio.h>
void input (int arr[], int n)
{
  int i;
  printf("Enter elements: /n ");
  for(i= 0; i<n; i++)
{
   scanf("\n %d", & arr[i]); }
}
  void display (int arr[], int n)
{
    int i;
    printf("Array elements enter by you:");
    for (i=0; i<n; i++)
{
    Printf("\n %d", arr [i]); }
}
  void minmax (int arr[], int n)
{
    int i, max, min;
    max = arr[0];
    for (i = 0; i<n; i++)
{
    if (arr [i] > max)
{
      max = arr[i];
}
}
```

```
  min = arr [0];
  for (i=0; i<n; i++)
{
   if (arr [i] < min)
{
    min = arr [i];
  }
}
  printf("\n largest number is = %d", max);
  printf("\n smallest number is = %d", min);
}

  int main()
{
  int arr[10], n, b [10], s [10];
  printf("Enter no. of elements: ");
  scanf("%d", &n);
  input (arr, n);
  display (arr, n);
  minmax (arr, n);

}
```

Date ___/___/_____

Output :-

```
Enter no. of elements : 5
Enter elements :
21
98
54
45
3

Array elements enter by you :
21
98
54
45
53

Largest number is = 98
Smallest number is = 3
```

Chitra
"It is the prime responsibility of every citizen to feel that his country." —Sardar Vallabh

Program = 6

```
// WAP to find odd and even element with
   the help of functions.
#include< stdio.h>
#include < conio.h>

void input (int arr[10], int n)
{
   int i;
   printf("Input array elements:");
   for( i=0; i<n; i++)
{
   Scanf("\n%d", &arr[i]);
}}
   void display ( int arr[10], int n)
{
   int i;
   printf("Array Elements Enter by you:");
   for (i=0; i<n; i++)
{
   printf("\n %d", arr[i]);
}}
   void even odd (int arr [10], int even [10], int
   odd [10], int n)
{
   int i, J=0, K=0;
   for( i=0; i<n ; i++)
{
   if( arr[i] %2 == 0)
{
   even [J] = arr [i];
   j++;
}
```

```
    else
    {
      odd [K] = arr [i];
      K++;
    }
  }

    printf("\n Even numbers are : ");
    for (i = 0; i < j ; i++)
  printf("\n %d", even [i]);
  printf("\n odd numbers are ");
    for (i = 0; i < K; i++)
    printf("\n %d", odd [i]); }

    int main()
  {
    int arr [10], n, even [10], odd [10];
  printf("Enter the no. of Elements : ");
  scanf("%d", &n);
  input (arr, n);
  display (arr, n);
  evenodd (arr, even, odd, n);

  }
```

Date ___/___/_____

Output:-

Enter the no. of elements : 5

Input array elements : 3
4
5
6
8

Even numbers are :
4
6
8

Odd numbers are :
3
5

Chitra
"Whenever you take a step forward, you are bound to disturb something." —Indira Gandhi

Program = 7

// WAP to add 10 in all the elements of an array with the help of functions.

```
# include <stdio.h>
# include <conio.h>
void input (int arr[ ], int n)
{
   int i;
   printf("Enter elements: \n");
   for( i= 0; i<n; i++)
{
   scanf("\n %d", &arr[i]);
}}
void display (int arr[], int n)
{
   int i;
   printf("Array elements enter by you: ");
   for (i=0; i<n; i++)
{
   scanf("\n %d", &arr[i]);
}}
void add ( int arr[], int n)
{
   int i;
   printf("\n Elements of an array after
          adding 10 : \n");
   for( i=0; i<n; i++)
{
   arr[i] = arr[i] + 10;
   printf ("\n %d", arr[i]);
}
}
```

```
int main()
{
    int arr[10], n;
    printf("Enter no. of elements:");
    scanf("%d", &n);
    input (arr, n);
    display (arr, n);
    add (arr, n);

}
```

Date ___/___/_____

Output :-

Enter no. of elements = 5

Enter elements:
2
3
4
5
6

Array elements enter by you:
2
3
4
5
6

Elements of an array after adding 10:
12
13
14
15
16

Chitra

"Whenever you take a step forward, you are bound to disturb something." —Indira Gandhi

Program 8

```
// WAP to show linear search by using function
#include <stdio.h>
void input (int arr[], int n);
void display (int arr[], int n);
void linearsearch(int arr[], int n, int target);
int main()
{
  int n, arr[10], target;
  printf("Enter the number of elements in the
          array:");
scanf("%d", &n);
printf("Enter the elements of the array: \n");
input (arr, n);
printf("The array is:");
display (arr, n);
printf("Enter the element to search for:"
scanf("%d", &target);
linearsearch (arr, n, target);
return 0;
}
void input (int arr[], int n)
{
  for (int i=0; i<n; i++)
{
  printf("Element [%d]: ", i);
  scanf("%d", &arr[i]);
}
}
void display (int arr[], int n)
{
```

```
{ for (int i = 0; i < n; i++)
  printf("%d", arr[i]);
}
  printf("\n");
}
void linearsearch (int arr[], int n, int target)
{
  int i;
for (i = 0; i < n; i++)
if (arr[i] = = target)
{
printf("Element %d found at Index %d
        \n", target, i);
}
}
```

Date __/__/____

Output:-

Enter the number of elements in the array:
5

Enter the elements of the array:
Element [0]: 7
Element [1]: 2
Element [2]: 21
Element [3]: 45
Element [4]: 17

The array is: 7 2 21 45 17
Enter the element to search for: 21
Element 21 found at Index 2

Chitra
"It is the prime responsibility of every citizen to feel that his country." —Sardar Vallabhai Patel

Program - 9

```
// WAP to show binary search by using functions
# include <stdio.h>
void inputArray ( int arr[ ], int size );
void display Array ( int arr[ ], int size);
int binarysearch (Int arr[ ], int size, int target);

int main()
{
int size, target, index;
printf (" Enter the number of elements in the
    array : ");
scanf ("%d", &size );
int arr [size ];
printf (" Enter the elements : \n");
input Array ( arr, size )
printf(" The array is : ");
display array ( Arr, size );
printf (" Enter the element to search for:");
scanf("%d", &target);
index = binarysearch (arr, size, target );
if (index != -1)
{
printf(" Element %d found at index %d.\n",
        target, index); }
else
{
printf( " Element not found \n "); }
return 0;
}
```

Date __/__/____

```
int binarysearch(int arr[], int size, int target)
{
    int low = 0;
    int high = size - 1;
    int mid;

  while (low <= high)
  {
    mid = (low+ high)/2;
    if (arr [mid] == target)
  {
      return mid;
  }
  else
  if (arr [mid ] < target )
  {
    low = mid+1;
  }
  else
  {
    high = mid-1;
  }
}
    return -1;  // target not found
}
  void input Array (int arr [], int size)
  {
    for ( int i = 0; i < size ; i++)
  {
    printf ("element [%d] : ", i);
    scanf ("%d", & arr [i]);
  }
```


"Whenever you take a step forward, you are bound to disturb something." —Indira Gandhi

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

Date ___/___/_____

Output :-

```
Enter the number of elements in the array:5

Enter the elements:
Element [0]: 3
Element [1]: 5
Element [2]: 7
Element [3]: 9
Element [4]: 12

The array is: 3 5 7 9 12
Enter the element to search for: 9
Element 9 found at index 3
```

Chitra

"Whenever you take a step forward, you are bound to disturb something." —Indira Gandhi

Program = 10

//WAP to show bubblesort by using function.

```
#include< stdio.h>
# include < conio.h>
void input (int arr[], int n);
void display (int arr[], int n);
void bubblesort (int arr[], int n);
void main()
{
int arr [10], n;
printf(" Enter the size of array: ");
scanf("%d", &n);
input (arr,n);
printf(" \n The array is \n");
display (arr, n);
printf("\n After the sorting array is \n");
bubblesort (arr,n);
display (arr,n);
getch();
}
void bubblesort (int arr[], int n)
{
int i, j, temp;
for (i=0; i<n-1; i++)
{
for (j= 0; j <n-1; j++)
{
if (arr[j] > arr[j+1])
{
temp = arr[j];
arr [j] = arr [j+1];
arr [j+1] = temp;
```

Chitra

"No religion has mandated killing others as a requirement for its sustenance or promotion." —Dr. A.P.J. Abdul Kalam

```
  }
 }
}}
 void input (int arr[], int n)
 {
   int i;
  for (i=0; i<n; i++)
 {
  printf ("Enter the array %d element", i);
  scanf ("%d", &arr[i]);
  }
 }
  void display (int arr[], int n)
 {
   int i;
   for (i=0; i<n; i++)
 {
   printf ("\n %d", arr[i]);
  }
 }
```

Date ___/___/_____

Output :-

```
Enter the size of array : 5
Enter the array [0] element 9
Enter the array [1] element 5
Enter the array [2] element 1
Enter the array [3] element 21
Enter the array [4] element 29

The array is 9 5 1 21 29
After the Sorting array is
1
5
9
21
29
```

Program = 11

```
// WAP for selection Sort

# include <stdio.h>
# include <conio.h>
void input (int arr[], int n);
void selectionsort (int arr[], int n);
void display (int arr[], int n);
void main()
{
  int arr[10], n;
  clrscr();
  printf("Enter the size of array");
  scanf("%d", &n);
  printf("Enter the array \n");
  input (arr, n);
  printf("\n The array is \n");
  display (arr, n);
  printf("\n After sorting the array is \n");
  selectionsort (arr, n);
  display (arr, n);
  getch();

}
  void selectionsort (int arr[], int n)
{
  int i, j, temp, min_index;
  for (i=0; i<n; i++)
{
  min_index = i;
  for (j = i+1; j<n; j++)
{
```

```
if (arr[j]<arr[min_index])
{
  min-index = j;
}
temp = arr[i];
arr[i] = arr[min_index];
arr[min_index] = temp;
}
}
void input (int arr[], int n)
{
  int i;
  for (i=0; i<n; i++)
{
  scanf("%d", &arr[i]);
}
  printf("\n");
}
void display (int arr[], int n)
{
  int i;
  for (i=0; i<n; i++)
{
  printf("%d", arr[i]);
}
}
```

Date ___/___/_____

Output :-

Enter the size of array : 5

Enter the array
99
66
55
21
10

The array is
99 66 55 21 10

After the sorting the array is
10 21 55 66 99

Chitra
"It is the prime responsibility of every citizen to feel that his country." —Sardar Vallabhai Patel

Program - 12

```
// WAP to show insertion sort

# include <stdio.h>
void insertionsort (int arr[], int n);
void printArray (int arr[], int n);
int main()
{
int arr[] = {12, 11, 13, 5, 6};
int n = size of (arr) / size of (arr[0]);
printf("Original array : \n");
printArray (arr, n);
insertionsort (arr, n);
printf("Sorted array : \n");
printArray (arr, n);
return 0;
}
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");
}
```

Date ___/___/_____

Output :-

Original array:

12 11 13 5 6

Sorted Array:

5 6 11 12 13

Program = 13

```
// WAP for addition of two matrices.
#include <stdio.h>
void input matrix (int matrix[10][10], int row,
                    int col);
void display matrix (int matrix[10][10], int row
                    int col);
void add matrices( int matrix1 [10][10], int
                    matrix 2 [10][10], int result [10]
                    [10], int row, int col);
int main()
{
  int row, col;
  int matrix1[10][10], matrix 2 [10][10], result [10]
                    [10];
printf (" Enter the number of rows and columns:")
scanf ("%d %d", &row, &col);
printf("Input for first matrix: \n");
input matrix (matrix 1, row, col);
printf ("Input for second matrix: \n");
input matrix (matrix 2, row, col);
add matrix (matrix 1, matrix 2, result, row,
                    col);
printf (" first matrix : \n");
  display matrix ( matrix 1, row, col);
printf("second matrix : \n");
display matrix (matrix 2, row, col);
printf (" Resultant matrix after addition : \n");
display matrix (result, row, col);
return 0;
}
```


"No religion has mandated killing others as a requirement for its sustenance or promotion." —Dr.A.P.J.Abdul Kalam

```
void Input matrix (int matrix [10][10], int row,
                   int col)
{
    printf(" Enter elements of the matrix: \n");
    for (int i=0; i<row; i++)
    {
      for (int j=0; j<col; j++)
      {
        printf(" Element [%d][%d]: ", i, j);
        scanf("%d", &matrix [i][j]);
      }
    }
}

void display matrix (int matrix [10][10],
                     int row, int col)
{
    for (int i=0; i<row; i++)
    {
      for (int j=0; j<col; j++)
      {
        printf("%d", matrix [i][j]);
      }
      printf("\n");
    }
}

void add matrices (int matrix 1 [10][10],
                   int matrix 2 [10][10], int result
                   [10][10], int row, int col)
{
    for (int i=0; i<row; i++)
    {
```

```
    for (int j=0; j<col; j++)
    {
        result[i][j] = matrix1[i][j] + matrix2
                       [i][j];
    }
}
}
```

Output:-

Enter the number of rows and columns: 2
2

Input for first matrix:
Element [0][0]: 2
Element [0][1]: 4
Element [1][0]: 6
Element [1][1]: 8

Input for second matrix:
enter elements of the matrix:
Element [0][0]: 1
Element [0][1]: 3
Element [1][0]: 5
Element [1][1]: 7

first matrix:
2 4
6 8

Second matrix:
1 3
5 7

Resultant matrix after addition:
3 7
11 15

Program = 14

```
//WAP to find transpose of a matrix
#include<stdio.h>
 void input matrix(int matrix[][10], int row, int col);
void display matrix (int matrix[][10], int row, int col);
void transpose matrix(int matrix[][10], int row, int col);
 int main()
{
 int row, col;
 int matrix[10][10], transposed[10][10];
 printf("Enter the number of rows and columns:");
 scanf("%d %d", &row, &col);
 printf("Input for matrix : \n");
  input matrix (matrix, row, col);
  transpose matrix (matrix, transposed, row, col);
  printf("Original matrix : \n");
  display matrix (transposed, col, row);
  return 0;
}
 void input matrix (int matrix[][10], int row,
                    int col)
{
 printf("Enter elements of the matrix : \n");
 for (int i = 0; i < row; i++)
{
  for ( int j = 0; j < col < j++)
{
   Printf ("Element [%d][%d]; ", i, j);
   Scanf ("%d", & matrix [i][j]);
  }
 }
}
```

```
void display matrix (int matrix[][10], int
                row, int col).
{
  for (int i=0; i< row; i++)
{
  for (int j=0; j< col; j++)
{
  printf("%d ", matrix[i][j]);
}
  printf("\n");
}
}
void transpose matrix (int matrix[][10], int
                transposed[][10], int row,
                int col)
{
  for (int i=0; i< row; i++)
{
  for (int j=0; j< col; j++)
{
  transposed[j][i]= matrix[i][j];
}
}
}
```

Date __/__/____

Output:

```
Enter the number of rows and columns : 2
2
Input for matrix:
Enter elements of the matrix:
Element [0][0] : 9
element [0][1] : 8
element [1][0] : 7
element [1][1] : 6

Original matrix:
9    8
7    6

Transposed matrix:
9    7
8    6
```

Chitra
"It is the prime responsibility of every citizen to feel that his country." —Sardar Vallabhai Patel

Program = 15

```
//WAP to multiply two matrices

# include <stdio.h>
void input (int matrix[][10], int row, int col);
void display (int matrix[][10], int row, int col);
void multiply (int matrix1[][10], int matrix2[][10],
        int result[][10], int row1, int col1, int row
        2, int col2);
int main()
{
  int row1, col1, row2, col2;
  int matrix[10][10], matrix2[10][10], result[10][10];
  printf("Enter the number of rows and columns for
        first matrix:");
  scanf("%d %d", &row1, &col1);
  printf("Enter the number of rows and columns
        for second matrix:");
  scanf("%d %d", &row2, &col2);
  printf("Input for first matrix :\n");
  input (matrix1, row1, col1);
  printf("Input for second matrix:\n");
  input (matrix2, row2, col2);
  multiply (matrix1, matrix2, result, row1, col1,
        row2, col2);
  printf("first matrix :\n");
  display (matrix1, row1, col1);
  printf("second matrix :\n");
  display (matrix2, row2, col2);
  printf("Resultant matrix after multiplication:
        \n");
  display (result, row1, col2);
```

Date __/__/____

```
return 0;
}
void input (int matrix[][10], int row, int col)
{
printf(" Enter elements of the matrix: \n");
for( int i=0; i< row; i++)
{
for (int j=0 ; j<col ; j++)
{
printf("Element [%d][%d]: ", i, j);
scanf("%d", &matrix[i][j]);
}
}
}
void display (int matrix [][10], int row, int col)
{
for (int i=0; i< row; i++)
{
for (int j=0; j<col; j++)
{
printf("%d", matrix [i][j]);
}
printf("\n");
}
}
void multiply ( int matrix1 [][10], int matrix2
[][10], int result [][10], int row1, int
col1, int row2, int col2)
{
for( int i=0; i<row1; i++)
}
```


"Whenever you take a step forward, you are bound to disturb something." —Indira Gandhi

```
{ for (int j=0; j< col2; j++)
      result [i][j] = 0;
  { for (int k=0; k < col1; k++)
      result [i][j] += matrix [i][k] * matrix 2
                                      [k][j];
    }
  }
}
```

Date __/__/____

Output =

Enter the number of rows and columns for first matrix : 2 2
Enter the ~~second~~ matrix of rows and columns for second matrix : 2 2

Input for first matrix :
Enter elements of the matrix :
Element [0][0] : 2
Element [0][1] : 4
Element [1][0] : 6
Element [1][1] : 8

Input for second matrix :
Enter elements of the matrix :
Element [0][0] : 1
Element [0][1] : 3
Element [1][0] : 5
Element [1][1] : 7

First matrix :
2 4
6 8
Second matrix :
1 3
5 7
Resultant matrix after multiplication :
22 34
47 74

```
// WAP of sparse array.
#include<stdio.h>
struct sparse
{
    int row;
    int col;
    int ele; };
void input( int mat[5][5], int r, int c);
void display (int mat[5][5], int r, int c);
void storesparse matrix( int mat[5][5], int r, int c,
                        structsparse sp[]);
void showsparseinfo( struct sparse sp[], int count);
void restorematrix (struct sparse sp[], int N[5][5]
                                        );

int main()
{
    int mat[5][5], r, c, N[5][5];
    struct sparse sp [25];
    printf("Enter no. of rows and columns");
    scanf("%d %d", &r, &c);
    input (mat, r, c);
    printf("\n The sparse matrix is: \n");
    display (mat, r, c);
    Storesparse matrix (mat, r, c, sp);
    Showsparseinfo ( sp, sp[0].ele);
    restore matrix (sp, N);
    printf("Restored matrix is : \n");
    display (N, r, c);
    return 0;
}
```

Date ___/___/___

```
void input (int mat [5][5], int r, int c)
{
    int i, j;
printf ("Enter the elements ");
for (i=0; i<r; i++)
{
  for (j=0; j<c; j++)
{
  scanf("%d", & mat[i][j]);
}
}}
void display (int mat[5][5], int r, int c)
{
    int i, j;
  for (i=0; i<r; i++)
{
  for (j=0; j<c; j++)
{
  printf("%d", mat[i][j]);
}
  printf("\n");
}}
void showsparseinfo (struct sparse sp[],
        int count)
{
    int i;
printf("sparse matrix Info :\n");
printf("Rows : %d, columns : %d, Non-zero
        elements: %d \n", sp[0].row, sp[0].
        col, sp[0].ele);
```

```
for (i=0; i<=count; i++)
{
  printf("Element %d: row= %d, col= %d, value=
          %d \n", i, sp[i].row, sp[i].col, sp[i].
          ele);
}

void storesparsematrix(int mat[s][s], int r, int
          c, struct sparse sp[])
{ int i, j, count=0;
  sp[0].row = r;
  sp[0].col = c;
  for(i=0; i<r; i++)
{ for( j=0; j<c; j++)
{ if(mat[i][j] != 0)
{ count++;
  sp[count].row=i;
  sp[count].col = j;
  sp[count].ele = mat[i][j]; }}}
  sp[0].ele = count; }
void restorespsmatrix(struct sparse sp[], int N
          [s][s])
{ int i, j, k=1;
  for (i=0; i<sp[0].row; i++)
{ for (j=0; j<sp[0].col; j++)
{ N[i][j]=0
}}
  for (i=1; i<=sp[0].ele; i++)
{
  N[sp[i].row][sp[i].col] = sp[i].ele;
  }
}
```

Output:

```
Enter no. of rows and columns 3
3
Enter the array 0
0
5
0
0
6
0
0
2
9
The sparse matrix is :
0 0 5
0 6 0
0 2 9
Sparse matrix info:
Rows : 3, Columns : 3 , Non-zero elements : 4
Element 0: row = 3, col = 3, value = 4
Element 1: row = 0, col = 2, value = 5
Element 2: row = 1, col = 1, value = 6
Element 3: row = 2, col = 1, value = 2
Element 4: row = 2, col = 2, value = 9

Restored matrix is:
0 0 5
0 6 0
0 2 9
```

Program 17

```
//WAP in C to show vector representation of lower
   triangular sparse matrix
# include <stdio.h>
# include <conio.h>
void storesparse( int sparse [ ]);
void restoresparse ( int sparse [ ]);
void main( )
{
  int sparse [7];
  storesparse (sparse);
  restoresparse (sparse);
   getch();
}
  void storesparse (int sparse [ ])
{
    int i, j, rc, K=2, count;
    clrscr();
  printf( "Enter no of rows & col ");
  scanf("%d", &rc);
  sparse [0] = rc;
   count = rc* (rc+1)/2;
    sparse [1] = count;
     for (i=0; i<rc; i++)
  {
     for (j=0; j<=i ; j++)
  {
   printf (" Enter [%d] [%d] element ", i, j);
  scanf ( "%d", &sparse [K]);
   K++;
}
}
}
```

```
void restoresparse(int sparse[])
{
  int i, j, K = 2;
  for (i = 0; i < sparse[0]; i++)
{
  for (j = 0; j < sparse[0]; j++)
{
  if (i >= j)
{
  printf("%d ", sparse[K]);
  K++;
}
  else
{
  printf("\n");
}
}
```

Date ___/___/_____ Page No.:

Output =

Enter no. of rows & col 3
Enter [0][0] element 1
Enter [1][0] element 2
Enter [1][1] element 3
Enter [2][0] element 4
Enter [2][1] element 5
Enter [2][2] element 6

1	0	0
2	3	0
4	5	6

Program - 18

//WAP to show vector representation of upper triangular matrix (sparse).

```
#include<stdio.h>
#include<conio.h>
void storesparse(int sparse[]);
void restoresparse (int sparse[]);
void main()
{
 int sparse[7];
 storesparse (sparse);
 restoresparse (sparse);
 getch();
}
 void storesparse(int sparse[])
{
  int i, j, rc, K=2, count;
  clrscr();
  printf("Enter no. of rows & col");
  scanf("%d", &rc);
 sparse[0] = rc;
 count = rc * (rc+1) / 2;
  Sparse[1] = count;
  for(i=0; i<rc; i++)
{
 for (j=i; j<=rc; j++)
{
 printf("Enter [%d][%d] element", i, j);
  scanf("%d", &sparse[K]);
  K++;
}}}
```

```
void restoresparse (int sparse[])
{
    int i, j, k = 2;
  for (i=0; i<sparse[0]; i++)
{
  for (j=0; j<sparse[0]; j++)
{
    if (i<=j)
  {
    printf("%d", sparse[k]);
     k++;
  }
  else
  {
   printf("0");
   }
  }
   printf("\n");
  }
  }
```

Date __/__/____

Output =

Enter no. of rows & col 3
Enter [0][0] element 2
Enter [0][1] element 3
Enter [0][2] element 4
Enter [0][3] element 5
Enter [1][1] element 6
Enter [1][2] element 7
Enter [1][3] element 8
Enter [2][2] element 9
Enter [2][3] element 11

2	3	4
0	5	6
0	0	7

"Whenever you take a step forward, you are bound to disturb something." —Indira Gandhi

Program = 19

```
//WAP to show vector representation of diagonal
 sparse matrix.
#include<stdio.h>
#include<conio.h>
void restaresparse (int sparse[]);
void staresparse (int sparse[]);
void main()
{
  int sparse[10];
  clrscr();
  starcsparse (sparse);
  restaresparse (sparse);
  getch();
}
void storesparse (int sparse[])
{
  int i, rc, count, K=2;
  printf(" Enter no. of rows and columns");
  scanf("%d", &rc);
  sparse[0] = rc;
  count = rc;
  sparse[1] = count;

  for (i=0; i<rc; i++)
{
  printf(" Enter [%d][%d] element", i, j);
  scanf("%d", &sparse[K]);
  K++;
}}
```

```
void restoresparse(int sparse[])
{
  int i, j, k = 2;
  printf("sparse matrix Info \n");
  printf("No. of Rows & columns = %d \n", sparse[0]);
  printf("No. of non zero element = %d \n", sparse[1]);
  for(i=0; i< sparse[0]; i++)
{
  for(j=0; j <sparse[0]; j++)
{
  if (i==j)
{
  printf("%d", sparse[k]);
  k++; }
  else
  printf("0");
}
  printf("\n");
}
}
```

Output =

Enter no. of rows and columns 3
Enter [0][0] element 7
Enter [1][1] element 6
Enter [2][2] element 8
Sparse matrix Info
no. of rows & columns = 3
No. of non zero element = 3

7	0	0
0	6	0
0	0	8

Program = 20

//WAP to show the vector representation of tri-diagonal sparse matrix.

```
# include<stdio.h>
# include<conio.h>
void storesparse (int sparse[]);
void restoresparse (int sparse[]);
void main()
{
 int sparse [15];
 storesparse (sparse);
 restoresparse (sparse);
 getch();
}
 void storesparse (int sparse[])
{
 int i, j, rc, K=2, count;
 clrscr();
 printf("enter no. of rows & col");
 scanf("%d", &rc);
 sparse [0] = rc;
 count = 3 * rc - 2;
 sparse [1] = count;
 for (i=0; i<rc; i++)
{
 for (j=0; j<rc; j++)
{
 if((i==j) :: (i==j+1) || (i==j-1))
{
 Printf("Enter [%d][%d] element", i, j);
 scanf("%d", &sparse [K]);
 K++;
```

```
}}
}}
void restoresparse (int sparse[])
{
int i, j, k=2;
for (i=0; i<sparse[0]; i++)
{
for (j=0; j<sparse[0]; j++)
{
if ((i==j) || (i==j+1) || (i==j-1))
{
printf("%d ", sparse[k]);
k++;
}
else
{
printf("0 ");
}
}
printf("\n");
}
}
```

Date __/__/____

Output =

Enter no. of rows & col 4
Enter [0][0] element 1
Enter [0][1] element 2
Enter [1][0] element 3
Enter [1][1] element 4
Enter [1][2] element 5
Enter [2][1] element 6
Enter [2][2] element 7
Enter [2][3] element 8
Enter [3][2] element 9
Enter [3][3] element 11

1	2	0	0
3	4	5	0
0	6	7	8
0	0	9	11

Program 21

//WAP to merge two sorted arrays.

```
#include<stdio.h>
void bubblesort (int arr[], int size);
void mergearrays (int arr1[], int size1, int
                arr2[], int size2, int result
                [] );
void display (int arr[], int size);
int main()
{
int size1, size2;
printf("Enter the size of 1st Array: ");
scanf("%d", &size1);
printf("Enter the size of 2nd Array: ");
scanf("%d", &size2);
int arr1[size1], arr2[size2];
int mergesize = size1 + size2;
int result[mergesize];
printf("Enter element of first array: \n");
for (int i=0; i<size1; i++)
{
scanf("%d", &arr1[i]);
}
printf("Enter element of second array: \n");
for (int i= 0; i<size2 ; i++)
{
scanf("%d", &arr2[i]);
}
bubblesort(arr1, size1);
bubblesort (arr2, size2);
```

```
merge arrays(arr1, size1, arr2, size2, result);
printf("\n merged sorted array:\n");
display (result, mergesize);
printf("\n");
return 0;
}
void bubblesort (int arr[], int size)
{
int temp;
for(int i=0; i<size-1; i++)
{
for(int j=0; j<size-1; j++)
{
if(arr[j] > arr[j+1])
{
temp = arr[j];
arr[j] = arr[j+1];
arr[j+1] = temp;
}
}}}
void display (int arr[], int size)
{
for(int i=0; i<size; i++)
{
printf("%d", arr[i]);
}
}
```

```
void mergearrays(int arr1[], int size1, int
  arr2[], int size2, int result[])
{
  int i=0; j=0; K=0;

  while (i<size1 && j< size2)
{
  if (arr1[i] <= arr2[j])
{
   result[K++] = arr1[i++] }

  else
{
   result[K++] = arr2[j++];
 }}
   while (i< size1)
{
   result [K++] = arr1[i++];
}
  while (j<size2)
{
   result [K++] = arr2[j++];
}
}
```

Date ___/___/___

Output

Enter the size of 1st array : 5
Enter the size of 2nd array : 6
Enter element of first array :
1
9
7
11
5
Enter element of second array :
5
8
1
12
7
14

After sorting 1st array is : 1 5 7 9 11
After sorting 2nd array is : 1 5 7 8 12 14
Merged Sorted array :
1 1 5 5 7 7 8 9 11 12 14

Program = 22

//WAP to reverse a string.

```
# include <stdio.h>
# include <conio.h>
# include <string.h>
# define Max20
# include <stdlib.h>

void push (char stack [20], int *top, char ch);
char pop (char stack [ ], int * top);
void main ()
{
 char string [20];
 char stack [MAX];
 int top = -1;
 int i;
 clrscr();
 printf(" Enter the string");
 gets (string);

 for (i = 0; i<strlen( string); i++)
{
  Push (stack, &top, string [i]);
}
 for (i=0; i < strlen(string); i++ )
{
  string [i] = pop(stack, &top);
}
  printf("\n Reverse string = %s", string);
}
  getch();
}
```

```
void push (char stack[], int * top, char c)
{
  if (*top == MAX-1)
{
  printf("\n The stack is full");
   exit(1);
  (*top)++;

  stack[*top] = c;
}
  char pop(char stack[], int *top)
{
  char item;
  if (*top == -1)
{
  printf("\n The stack is empty");

   exit(1);
}
  item = stack[*top];
  (*top)--;
  return item;

}
```

Page N

Date ___/___/_____

Output:-

Enter the string Abhinav

Reverse string = vanihbA

//WAP to show the use and representation of stack.

```
#include<stdio.h>
#include<conio.h>
# define MAX 100

void push( int stack[], int* top, int value);
int pop( int stack[], int* top);
void display ( int stack[], int top);

int main ()
{
 int stack[MAX];
 int top = -1;
 int choice, value;
 while (1){
 printf("\n Stack Menu: \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:
 if (top >= MAX-1)
 {
   printf (" Stack overflow \n");
 }
```

Date __/__/____

```
else {
 printf("Enter the value to push: ");
 scanf("%d", &value);
 push(stack, &top, value);
}
 break;

 Case 2:
  if(top==-1)
{ printf("Stack underflow \n");
}
   else{
    value = pop(stack, &top);
    printf("Popped value: %d \n", value);
}
  break;

 Case 3:
  if(top==-1){
  printf("stack is empty \n");
}
  else{
   display(stack, top);
}
    break;
 case 4:
  printf("Existing.... \n");
   exit(0);
```

```
default:
 printf("Invalid choice: \n"); }}
  return 0;
}
void push (int stack[], int *top)
{
  int value = stack[*top];
  *top = *top-1;
  return value;
}
void pop (int stack[], int *top)
{
  int value = stack[*top];
  *top = *top-1;
  return value;
}
void display (int stack[], int top)
{
  printf("stack elements are: \n");
  for (int i = top; i >= 0; i--)
{
  printf("%d \n", stack[i]);
}
}
```

Date ___/___/___

Output -

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

Stack Menu:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to push: 3
pushed 3 to stack

Stack menu:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice : 3
Stack elements are
5
3

Program 24

//WAP to show peek function of stack.

```
# include<stdio.h>
# include <conio.h>
# include <stdlib.h>
# define MAX 100
void push (int stack[], int *top, int item);
int pop(int stack[], int *top);
int peek (int stack[], int* top);

void main()
{
  int stack [MAX];
  int item, ch, top = -1;

  do {
  printf("\n 1. Push");
  printf("\n 2. pop");
  printf("\n 3. peek");
  printf("\n 4. Exit");
  Printf("\n Enter your choice:");
  Scanf("%d", &ch);

  Switch(ch)
{
Case 1: printf("Enter item to insert");
        scanf("%d", &item);
        push(stack, &top, item);
        break;

Case 2: item = pop (stack, &top)
        printf("Deleted item = %d", item);
        break;
```

Chitra
"No real change in history has ever been acheived by discussions." —Subhash chandra bose

Date __/__/____

```
Case 3: item = peek (stack, &top);
        printf("Item at the top = %d", item);
        break;

Case 4: exit(1);
 default: printf("Wrong choice");
 }
    while (ch);
}
  void push(int stack[], int *top, int item)
{
  if (stack[*top] >= MAX-1)
{
    printf("Stack is full");
     exit (1);
}

  else
{
   (*top)++;
   stack[*top] = item; }
}
   int pop(int stack[], int *top)
{
    int item;
  if (stack[*top] == -1)
{
    printf(" Stack empty");
    exit (1);
}
    item = Stack[*top];
    (*top)--;
```

```
    return item;
}
int peek(int stack[], int * top)
{
    int item;
    if (* top = = -1)
{
    printf("\n stack Empty");
    exit(1);
}
    item = stack [ * top];
    return item;
}
```

Date / /

Output:

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

Enter your choice : 1
Enter item to insert 5

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

Enter your choice : 1
Enter item to insert 6

1. Push
2. Pop
3. Peek

Enter your choice : 2
deleted item = 6

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

Enter your choice : 3
item at the top = 5

Program 25

```
//WAP to check validity of brackets in the
   expression.
#include <stdlib.h>
#include <conio.h>
# include <string.h>
# include <stdlib.h>
#  define MAX 15

int check (char exp[] );
int match( char a, char b);
void push( char[], int *, char);
char pop( char [], int * );

 void main()
{
  char exp[15];
   int valid;
  printf(" Enter an expression");
  gets (exp);
  valid = check (exp);
  if (valid = = 1)
 {
    printf(" Expression is valid ");
 }
   else
  printf("\n Not valid");
 }
 else
{
printf("\n left paranthesis are more than
      right paranthesis " );
```

Chitra "Religion has mandated killing others as a requirement for its sustenance or promotion." —Dr.A.P.J.Abdul Kalam

```
int check( char exp[ ])
{
 int i;
 char item;
 char stack [MAX];
 int top = -1;
 for ( i=0; i < strlen (exp); i++)
{
 if (exp[i] == '(' || exp[i] == '[' || exp[i] ==
  '{')
 push(stack, &top, exp[i] );
 if (exp[i] == ')' || exp[i] == ']' || exp[i] == '}')
{
 if (top == -1)
{
 printf("\n Right paranthesis are more than left
  paranthesis");
 return 0;
}
 else
{
 item = pop(stack, &top);
 if (!match (item, exp[i]))
{
 printf("\n Paranthesis is Matched");
  return 0;
}}}}
 if (top == -1)
{
 printf("\n Balanced Parenthesis \n");
 return 1;
}
```

Date __/__/____

```
return 0; }}
int match (char a, char b)
{
    if (a == '[' && b == '[')
        return 1;
    if (a == '{' && b == '{')
        return 1;
    if (a == '(' && b == ')')
        return 1;
        return 0;

}

void push(char stack[], int * top, char item)
{
  if (* top == MAX-1)
{
  printf ("\n Stack overflow");
    exit (1); }
  (* top) ++;
  stack[* top] = item; }

Char pop (char stack[], int * top)
{
    if (* top == -1)
{
  printf("\n Stack Empty");
    exit (1);
}
    return (stack[(* top) --]);

}
```

Date ___/___/_____

Output:-

Enter an expression (a+b)* d & +e
Left paranthesis are more than right paranthesis
Not Valid

```
// WAP to find the factorial with the help of recursion.
# include <stdio.h>
# include <conio.h>
long int factorial(int);
void main()
{
 int num, fact;
 printf("enter no.");
 scanf("%d", &num);
 fact = factorial(num);
 printf("\n factorial= %d", fact);
 getch();
}
 long int factorial(int n)
{
 if (n==0)
 return 1;
 else
 return (n*factorial(n-1));
}
```

Page No.:

Date ___/___/_____

Output :-

enter no. 3

Factorial = 6

Program 27

```
// WAP to find the fibonacci series in DS.
#include<stdio.h>
#include<conio.h>
 int fib (int i);
  void main()
{
  int term, i;
  clrscr();
  printf("enter no of term");
  scanf("%d\n", &term);
  for(i=0; i<term; i++)
{
  printf("%d", fib(i));
}
 getch();
}
 int fib (int i)
{
  if (i==0)
   return 0;
   else
   if (i==1)
    return 1;
     else
   return (fib(i-1) + fib(i-2));
}
```

Date ___/___/_____

Page No.

Output :-

enter no of term 4
4
0 1 1 2

Program 28

```
// WAP to perform the queue & Dequeue operation.
#Include <stdio.h>
#Include< conio.h>
# include < process.h>
# define ms
void enqueue (int queue [m], int* f , int* r, int
                item);
void dequeue (int queue[m], int* f , int* r);
void peek (int queue [m], int* f, int* r);
void main()
{
    int queue [m], f, r, ch ,item;
    clrscr();
    printf("%. enqueue \n2. dequeue \n3. peek \n4.
                exit ");
    f = -1;
    r = -1;
    do
    {
    printf("\n Enter the choice = ");
    Scanf(" %d ", &ch);
    Switch (ch)
    {
    case 1:
    enqueue (" queue, &f, &r, item);
    break;
    case 2:
    dequeue (queue , &f , &r);
    break;
    case 3:
    peek (queue , &f, &r);
    break;
```


"No real change in history has ever been acheived by discussions." —Subhash chandra bose

Date __/__/____

```
case 4:
 exit(1);
 break;
}}
while(ch);
 getch();
}
void enqueue (int queue[m], int *f, int *r, int item)
{
 if ((*f == 0 && *r == m-1) || (*f == *r+1))
{
  printf("\n overflow");
  return;
 }
 Printf("Enter element = ");
 Scanf("%d", &item);
  if (*f == -1)
{
  *f = 0;
  *r = 0;
 }
  else if (*r == m-1)
 *r = 0;
  else
 *r = *r+1;
 queue[*r] = item;
}
void dequeue (int queue[m], int *f, int *r
{
 int item;
```


"Whenever you take a step forward, you are bound to disturb something." —Indira Gandhi

```
if (*f == -1)
{
  printf("\n underflow");
  return;
}
item= queue[*f];
printf("deleted item = %d", item);
if (*f == *r)
{
*f = -1, *r = -1;
}
else if (*f == m-1)
*f = 0;
else
*f = *f+1;
}
void peek (int queue[m], int *f, int *r)
{
int item;
if (*f == -1)
{
printf("\n underflow");
return;
}
printf("peek item is = %d", queue[*r])
}
```

Date ___/___/_____ Page No.:

Output:-

1. enqueue
2. dequeue
3. peek
4. exit

Enter the choice = 2
underflow
Enter the choice = 3
underflow
Enter the choice = 1
Enter element = 2
Enter the choice = 1
Enter element = 4
Enter the choice = 1
Enter the ~~cho~~ element = 6
enter the choice = 1
Enter element = 8
Enter the choice = 3

Program 29

```
//WAP to Insert the last node in linked list.
#Include<stdib.h>
#Include<conio.h>
#include<stdlib.h>
typedef struct node
{
  int Info;
 struct node * link;
}
  node;
  void main()
{
  int i, n;
  node * start, * current, * temp;
  start = (node*) malloc (size of (node));
  current= start;
  printf(" how many nodes you want \n");
  scanf("%d", &n);
  for (i=1; i<=n; i++)
{
  printf("enter the information of %d th node" i);
 scanf("%d", &current -> info );
  if (i<n)
{
  current -> link = (node*) malloc (size of (node
                                    ));
  current= current -> link;
}
  else.
  Current -> link = NULL;
}
```

Date __/__/____

```
current = start;
while ( current! = NULL)
{
  printf(" %d", current -> info);
  current = current -> link;
}
  current = start;
  while (current! = NULL)
  {
    temp = current;
    current = current -> link;
    free ( temp);
  }
  getch();
}
```

Output :-

how many nodes you want
5
enter the information of ith node :
2
3
4
5
6

After Inserting 1 at the end :
2
3
4
5
6
1