Assignment 7

1. read n number of values in an array and display it in reverse order.

```
#include<stdio.h>
int main()
{
   int arr[20],i,size;
   printf("Enter the size : ");
   scanf("%d",&size);
   printf("Enter %d number of elements : ",size);
   for(i=0; i<size; i++)
      scanf("%d",&arr[i]);
   printf("Array elements in reverse order :\n");
   for(i=size-1; i>=0; i--)
      printf("%d ",arr[i]);

   getch();
   return 0;
}
```

2. find the sum of all elements of the array.

```
#include<stdio.h>
int main()
{
   int arr[20],i,size,sum=0;
   printf("Enter the size : ");
   scanf("%d",&size);
   printf("Enter %d number of elements : ",size);
   for(i=0; i<size; i++)
   {
      scanf("%d",&arr[i]);
      sum=sum+arr[i];
   }
   printf("Sum of array elements is %d",sum);
   getch();
   return 0;
}
```

3. copy the elements of one array into another array.

```
#include<stdio.h>
int main()
{
   int arr[20],res[20],i,size;
   printf("Enter the size : ");
   scanf("%d",&size);
   printf("Enter %d number of elements : ",size);
   for(i=0; i<size; i++)
      scanf("%d",&arr[i]);
   for(i=0; i<size; i++)
      res[i]=arr[i];
   printf("Copy array is :\n");
   for(i=0; i<size; i++)
      printf("%d ",res[i]);

   getch();
   return 0;
}
```

4. count a total number of duplicate elements in an array.

```
#include<stdio.h>
int main()
{
   int arr[20],dup[20],i,size,d=1,c=0,j,nOd=0,k;
   printf("Enter the size : ");
   scanf("%d",&size);
   printf("Enter %d number of elements : ",size);
   for(i=0; i<size; i++)
      scanf("%d",&arr[i]);
   for(i=0; i<size; i++)
      dup[i]=0;
   // CODE TO FIND DUPLICACY
   for(i=0; i<size; i++)
   {
      d=1;
      for(j=i+1; j<size; j++)
      {
         for(k=0; k<nOd; k++)
         {
            if(j==dup[k])
               break;
         }
```

```
            if(k!=nOd)
                continue;
            if(arr[i]==arr[j])
            {
                dup[c]=j;
                d++;
                c++;
                nOd++;
            }
        }
        for(k=0; k<nOd; k++)
        {
            if(i==dup[k])
                break;
        }
        if(k==nOd)
            printf("%d duplicacy is %d\n",arr[i],d);
    }

    getch();
    return 0;
}
```

5. find the maximum and minimum element in an array.

```
#include<stdio.h>
int main()
{
   int arr[20],i,size,max,min;
   printf("Enter the size : ");
   scanf("%d",&size);
   printf("Enter %d number of elements : ",size);
   for(i=0; i<size; i++)
      scanf("%d",&arr[i]);

   max=min=arr[0];
   for(i=1; i<size; i++)
   {
      if(max<arr[i])
         max=arr[i];
      else if(min>arr[i])
         min=arr[i];
   }
   printf("Max = %d Min = %d",max,min);
   getch();
   return 0;
}
```

6. separate odd and even integers in separate arrays.

```
#include<stdio.h>
int main()
{
   int arr[20],odd[20],even[20],i,size,j,k;
   printf("Enter the size : ");
   scanf("%d",&size);
   printf("Enter %d number of elements : ",size);
   for(i=0; i<size; i++)
      scanf("%d",&arr[i]);
   for(i=0,j=0,k=0; i<size; i++)
   {
      if(arr[i]%2==0)
      {
         even[j]=arr[i];
         j++;
      }
      else
      {
         odd[k]=arr[i];
         k++;
      }
   }
   printf("Array elements in even :\n");
   for(i=0; i<j; i++)
      printf("%d ",even[i]);
   printf("\nArray elements in odd :\n");
   for(i=0; i<k; i++)
      printf("%d ",odd[i]);
   getch();  return 0; }
```

7. insert New value in the array.

```
#include<stdio.h>
int main()
{
   int arr[10]={11,22,33,44,55,66},i,n,loc;
   printf("Array elements are : \n");
   for(i=0; i<6; i++)
      printf("%d ",arr[i]);
   printf("\nEnter the new value and location : ");
   scanf("%d%d",&n,&loc);
   if(loc<=6)
   {
      for(i=5; i>=loc; i--)
      {
         arr[i+1]=arr[i];
      }
      arr[loc]=n;
      printf("\nNew array is \n");
      for(i=0; i<7; i++)
      printf("%d ",arr[i]);
   }
   else
      printf("Try location between 0 to 5 : ");
   getch();
   return 0;
}
```

8. delete an element at desired position from an array.

```
#include<stdio.h>
int main()
{
   int arr[10]={11,22,33,44,55,66},i,n,dloc;
   printf("Array elements are : \n");
   for(i=0; i<6; i++)
      printf("%d ",arr[i]);
   printf("\nEnter the delete location : ");
   scanf("%d",&dloc);
   if(dloc<=6)
   {
      for(i=dloc; i<6; i++)
      {
         arr[i]=arr[i+1];
      }
      printf("\nNew array is \n");
      for(i=0; i<5; i++)
         printf("%d ",arr[i]);
   }
   else
      printf("Try location between 0 to 5 : ");
   getch();
   return 0; }
```

9. find the second largest element in an array.

```
#include<stdio.h>
int main()
{
   int arr[10],i,fLarg,sLarg,size;
   printf("Enter size : ");
   scanf("%d",&size);
   printf("%d elements : ",size);
   for(i=0; i<size; i++)
      scanf("%d",&arr[i]);
   fLarg=sLarg=arr[0];
   for(i=0; i<size; i++)
   {
      if(fLarg<arr[i])
      {
         sLarg=fLarg;
         fLarg=arr[i];
      }
      else if(sLarg<arr[i] && arr[i]!=fLarg)
         sLarg=arr[i];
   }
   printf("Second Larges : %d ",sLarg);
   getch();
   return 0; }
```

10. find the median of two sorted arrays of same size.

```
#include<stdio.h>
int main()
{
   int
arr1[5]={10,20,30,40,50},arr2[5]={11,22,33,44,55};
   int i,median,merge[10],round,temp,j;
   for(i=0; i<5; i++)
   {
        merge[i]=arr1[i];
   }
   for(j=0; j<5; j++,i++)
   {
        merge[i]=arr2[j];
   }
   printf("\nmerge array is : \n");
   for(i=0; i<10; i++)
     printf("%d ",merge[i]);
```

```
    for(round=1; round<=4; round++)
    {
        for(i=0; i<10-round; i++)
        {
            if(merge[i]>merge[i+1])
            {
                temp=merge[i];
                merge[i]=merge[i+1];
                merge[i+1]=temp;
            }
        }
    }
    printf("\nmerge array is : \n");
    for(i=0; i<10; i++)
        printf("%d ",merge[i]);
    printf("\nMedian is %d",(merge[5]+merge[5-1])/2);
    getch();
    return 0;
}
```

11. multiplication of two square Matrices.

```
#include<stdio.h>
int main()
{
   int arr1[5][5],arr2[5][5],mul[5][5];
   int i,j,k,size,sum=0;
   printf("Enter the size : ");
   scanf("%d",&size);
   printf("Enter 1st %d * %d matrix :\n",size,size);
   for(i=0; i<size; i++)
      for(j=0; j<size; j++)
         scanf("%d",&arr1[i][j]);
   printf("\nEnter 2nd %d * %d matrix :\n",size,size);
   for(i=0; i<size; i++)
      for(j=0; j<size; j++)
         scanf("%d",&arr2[i][j]);
   printf("\nthe 1st matrix is : \n");
   for(i=0; i<size; i++)
   {
      for(j=0; j<size; j++)
         printf("%d ",arr1[i][j]);
      printf("\n");
   }
   printf("\nthe 2nd matrix is : \n");
   for(i=0; i<size; i++)
   {
        for(j=0; j<size; j++)
            printf("%d ",arr2[i][j]);
        printf("\n");
    }

    for(i=0; i<size; i++)
    {
        for(j=0; j<size; j++)
        {
            sum=0;
            for(k=0; k<size; k++)
            {
                sum=sum+arr1[i][k]*arr2[k][j];
            }
            mul[i][j]=sum;
        }
    }
    printf("\nmultiplication matrix matrix is : \n");
    for(i=0; i<size; i++)
    {
        for(j=0; j<size; j++)
            printf("%d ",mul[i][j]);
        printf("\n");
    }
}
```

12. find transpose of a given matrix.

```
#include<stdio.h>
int main()
{
    int arr[5][5],i,j,trp[5][5],size;
    printf("Enter the size : ");
    scanf("%d",&size);
    printf("Enter %d*%d elements : \n",size,size);
    for(i=0; i<size; i++)
        for(j=0; j<size; j++)
            scanf("%d",&arr[i][j]);
    printf("\nthe matrix is : \n");
    for(i=0; i<size; i++)
    {
        for(j=0; j<size; j++)
            printf("%d ",arr[i][j]);
        printf("\n");
    }
    for(i=0; i<size; i++)
        for(j=0; j<size; j++)
            trp[j][i]=arr[i][j];
    printf("\nthe transpose matrix is : \n");
    for(i=0; i<size; i++)
    {
        for(j=0; j<size; j++)
            printf("%d ",trp[i][j]);
        printf("\n");
    }
}
```

13. find the sum of left diagonals of a matrix.

```
#include<stdio.h>

int main()
{
    int arr[5][5],i,j,trp[5][5],size,sum=0;
    printf("Enter the size : ");
    scanf("%d",&size);
    printf("Enter %d*%d elements : \n",size,size);
    for(i=0; i<size; i++)
        for(j=0; j<size; j++)
            scanf("%d",&arr[i][j]);
    printf("\nthe matrix is : \n");
    for(i=0; i<size; i++)
    {
        for(j=0; j<size; j++)
            printf("%d ",arr[i][j]);
        printf("\n");
    }
    for(i=0; i<size; i++)
    {
        sum=sum+arr[i][i];
    }
    printf("Sum of left diagonal is %d",sum);
    getch();
    return 0;
}
```

14. check whether a given matrix is an identity matrix.

```
#include<stdio.h>
int main()
{
   int arr[5][5],i,j,size,k;
   printf("Enter the size : ");
   scanf("%d",&size);
   printf("Enter %d*%d elements : \n",size,size);
   for(i=0; i<size; i++)
      for(j=0; j<size; j++)
         scanf("%d",&arr[i][j]);
   printf("\nthe matrix is : \n");
   for(i=0; i<size; i++)
   {
      for(j=0; j<size; j++)
         printf("%d ",arr[i][j]);
      printf("\n");
   }
   for(k=0; k<size; k++)
   {
      if(arr[k][k]!=1)
         break;
   }
   if(k!=size)
```

```
        printf("NOT a identity matrix.");
    else
    {
        for(i=0; i<size; i++)
        {
            for(j=0; j<size; j++)
            {
                if(i==j)
                    continue;
                if(arr[i][j]!=0)
                    break;
            }
            if(j!=size)
                break;
        }
        if(i!=size)
            printf("NOT a identity matrix 1.");
        else
            printf("IDENTITY.");
    }
    getch();
    return 0;
}
```

15. search an element in a row wise and column wise sorted matrix.

```
int main()
{
   int arr1[5][5],temp,round,element;
   int i,j,size;
   printf("Enter the size : ");
   scanf("%d",&size);
   printf("Enter 1st %d * %d matrix :\n",size,size);
   for(i=0; i<size; i++)
      for(j=0; j<size; j++)
         scanf("%d",&arr1[i][j]);
   printf("\nthe 1st matrix is : \n");
   for(i=0; i<size; i++)
   {
      for(j=0; j<size; j++)
         printf("%d ",arr1[i][j]);
      printf("\n");
   }
   // row wise sorting...
   for(i=0; i<size; i++)
```

```
{
   for(round=1; round<size; round++)
   {
      for(j=0; j<size-round; j++)
      {
         if(arr1[i][j]>arr1[i][j+1])
         {
            temp=arr1[i][j];
            arr1[i][j]=arr1[i][j+1];
            arr1[i][j+1]=temp;
         }
      }
   }
}
printf("\nthe Sorted Wise matrix is : \n");
for(i=0; i<size; i++)
{
   for(j=0; j<size; j++)
      printf("%d ",arr1[i][j]);
   printf("\n");
}
// column wise sorting...
```

```
for(i=0; i<size; i++)
{
   for(round=1; round<size; round++)
   {
      for(j=0; j<size-round; j++)
      {
         if(arr1[j][i]>arr1[j+1][i])
         {
            temp=arr1[j][i];
            arr1[j][i]=arr1[j+1][i];
            arr1[j+1][i]=temp;
         }
      }
   }
}
printf("\nthe Sorted Wise matrix is : \n");
for(i=0; i<size; i++)
{
   for(j=0; j<size; j++)
      printf("%d ",arr1[i][j]);
   printf("\n");
}
```

```
    printf("Which element you want to search : ");
    scanf("%d",&element);
    for(i=0; i<size; i++)
    {
        for(j=0; j<size; j++)
        {
            if(arr1[i][j]==element)
                break;
        }
        if(j!=size)
            break;
    }
    if(i!=size)
        printf("%d is found at index arr1[%d][%d]",arr1[i][j],i,j);
    else
        printf("%d is not found",element);
    getch();
    return 0;
}
```