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

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

return 0;
}
```

Output:

```
enter size of array :: 5
Input number :: 3
Input number :: 6
Input number :: 4
Input number :: 8
Input number :: 9
array in actual order :: 3 6 4 8 9
array in reverse order :: 9 8 4 6 3
```

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

```
#include <stdio.h>
int main(){
int n,i,sum=0;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n];
for(i=0;i<n;i++){
   printf("Input number :: ");
   scanf("%d",&arr[i]);
}

for(i=0;i<n;i++){
   sum += arr[i];
}
printf("Sum of all elements of the array is %d",sum);

return 0;
}
```

Output:

```
enter size of array :: 4
Input number :: 5
Input number :: 5
Input number :: 5
Input number :: 5
Sum of all elements of the array is 20
```

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

```
#include <stdio.h>
int main(){
int n,i,sum=0;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n],arrcopy[n];
for(i=0;i<n;i++){
   printf("Input number in first array :: ");
   scanf("%d",&arr[i]);
}

for(i=0;i<n;i++){
   arrcopy[i] = arr[i];
}
printf("Elements copy in second array :: ");
for(i=0;i<n;i++){
   printf("%d ",arrcopy[i]);
}

return 0;
}
```

Output:

```
enter size of array :: 4
Input number in first array :: 2
Input number in first array :: 5
Input number in first array :: 3
Input number in first array :: 6
Elements copy in second array :: 2 5 3 6
```

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

```
#include <stdio.h>
int main(){
int n,i,j,c=0;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n];
for(i=0;i<n;i++){
   printf("Input number in array :: ");
   scanf("%d",&arr[i]);
}

for(i=0;i<n;i++){
   j=i;
   for(j+=1;j<n;j++){
      if(arr[i]==arr[j]){
         c ++;
         break;
      }

   }

}

printf("Here total %d duplicate elements",c);

return 0;
}
```

Output:

```
enter size of array :: 6
Input number in array :: 4
Input number in array :: 4
```

Input number in array :: 2
Input number in array :: 2
Input number in array :: 3
Input number in array :: 3
Here total 3 duplicate elements

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

```
#include <stdio.h>
int main(){
int n,i,j,temp=0;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n];
for(i=0;i<n;i++){
   printf("Input number in array :: ");
   scanf("%d",&arr[i]);
}

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
      if(arr[i] < arr[j]){
         temp = arr[i];
         arr[i] = arr[j];
         arr[j] = temp;
      }
   }

}
printf("minimum element = %d \nmaximum element = %d",arr[0],arr[n-
1]);
return 0;
}
```

Output:

```
enter size of array :: 5
Input number in array :: 3
Input number in array :: 2
Input number in array :: 5
Input number in array :: 6
Input number in array :: 4
```

minimum element = 2
maximum element = 6

5. separate odd and even integers in separate arrays.

```
#include <stdio.h>
int main(){
int n,i,j=0,k=0,c=0;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n],odd[n],even[n];
for(i=0;i<n;i++){
   printf("Input number in array :: ");
   scanf("%d",&arr[i]);
}

for(i=0;i<n;i++){
 if(arr[i] % 2 == 0){
    even[j] = arr[i];
    j +=1;
    c += 1;
 }
 else{
    odd[k] = arr[i];
    k += 1;
 }
}
printf("Even numbers :: ");
for(i=0;i<c;i++){
   printf("%d ",even[i]);
}

printf("\nodd numbers :: ");
for(i=0;i<n-c;i++){
   printf("%d ",odd[i]);
}
```

```
return 0;
}
```

Output:

```
enter size of array :: 5
Input number in array :: 1
Input number in array :: 2
Input number in array :: 3
Input number in array :: 4
Input number in array :: 5
Even numbers :: 2 4
odd numbers :: 1 3 5
```

6. insert New value in the array.

```
#include <stdio.h>
int main(){
int n,m,i;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n],new[n];
for(i=0;i<n;i++){
   printf("Input number in array :: ");
   scanf("%d",&arr[i]);
}
printf("Inserted elements :: ");
for(i=0;i<n;i++){
   printf("%d ",arr[i]);
}

printf("\nhow many new elements you want to insert :: ");
scanf("%d",&m);
arr[n+m];

for(i=n;i<n+m;i++){
   printf("\ninput new number :: ");
   scanf("%d",&arr[i]);
}

printf("\nArray with New inserted elements :: ");

for(i=0;i<n+m;i++){
   printf("%d ",arr[i]);
}

return 0;
}
```

Output:

```
enter size of array :: 5
Input number in array :: 1
Input number in array :: 5
Input number in array :: 4
Input number in array :: 6
Input number in array :: 3
Inserted elements :: 1 5 4 6 3
how many new elements you want to insert :: 3

input new number :: 9

input new number :: 10

input new number :: 13

Array with New inserted elements :: 1 5 4 6 3 9 10 13
```

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

```
#include <stdio.h>
int main(){
int n,m,i,temp;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n];
for(i=0;i<n;i++){
   printf("Input number in array :: ");
   scanf("%d",&arr[i]);
}
printf("Inserted elements :: ");
for(i=0;i<n;i++){
   printf("%d ",arr[i]);
}

printf("\nEnter index for delete a number :: ");
scanf("%d",&m);

for(i=m;i<n;i++){
   arr[m] = arr[m+1];
   m = m + 1;
}

for(i=0;i<n-1;i++){
   printf("%d ",arr[i]);
}

return 0;
}
```

Output:

```
enter size of array :: 6
Input number in array :: 1
Input number in array :: 2
Input number in array :: 3
Input number in array :: 4
Input number in array :: 5
Input number in array :: 6
Inserted elements :: 1 2 3 4 5 6
Enter index for delete a number :: 5
1 2 3 4 5
```

9. find the second largest element in an array.

```
#include <stdio.h>
int main(){
int n,i,j,temp=0;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n];
for(i=0;i<n;i++){
   printf("Input number in array :: ");
   scanf("%d",&arr[i]);
}

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
      if(arr[i] < arr[j]){
         temp = arr[i];
         arr[i] = arr[j];
         arr[j] = temp;
      }
   }

}
printf("Second largest value = %d",arr[n-2]);
return 0;
}
```

Output:

```
enter size of array :: 6
Input number in array :: 3
Input number in array :: 4
Input number in array :: 8
Input number in array :: 9
Input number in array :: 7
Input number in array :: 2
Second largest value = 8
```

11. multiplication of two square Matrices.

```
#include <stdio.h>
int main(){
int n,i=0,j=0,k;
printf("enter size of array :: ");
scanf("%d",&n);
int arr1[n][n],arr2[n][n],mul[n][n];

printf("Enter elements in first array -->\n");
for(i=0;i<n;i++){
   for(j=0;j<n;j++){
       printf("Enter a number :: ");
      scanf("%d",&arr1[i][j]);

   }
}

printf("Enter elements in second array -->\n");
for(i=0;i<n;i++){
   for(j=0;j<n;j++){
       printf("Enter a number :: ");
      scanf("%d",&arr2[i][j]);

   }
}

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
       mul[i][j]=0;
       for(k=0;k<n;k++){
       mul[i][j] += arr1[i][k] * arr2[k][j] ;
```

```
      }

   }
}
printf("Multiplication of the given matrix --> \n");
for(i=0;i<n;i++){
   for(j=0;j<n;j++){
      printf("%d ",mul[i][j]);
   }
   printf("\n");
}
return 0;
}
```

Output:

```
enter size of array :: 2
Enter elements in first array -->
Enter a number :: 4
Enter a number :: 2
Enter a number :: 2
Enter a number :: 4
Enter elements in second array -->
Enter a number :: 2
Enter a number :: 2
Enter a number :: 2
Enter a number :: 2
Multiplication of the given matrix -->
12 12
12 12
```

12. find transpose of a given matrix.

```
#include <stdio.h>
int main(){
int n,i=0,j=0,k;
printf("enter size of array :: ");
scanf("%d",&n);
int arr1[n][n],arr2[n][n];

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
       printf("Enter a number :: ");
      scanf("%d",&arr1[i][j]);

   }
}

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
       arr2[i][j] = arr1[j][i];
   }
}

printf("Inserted matrix -->\n");
for(i=0;i<n;i++){
   for(j=0;j<n;j++){
       printf("%d ",arr1[i][j]);
     }
     printf("\n");
}

printf("Transpose of this given matrix is -->\n");
for(i=0;i<n;i++){
   for(j=0;j<n;j++){
       printf("%d ",arr2[i][j]);
     }
     printf("\n");
```

```
}
return 0;
}
```

Output:

```
enter size of array :: 3
Enter a number :: 1
Enter a number :: 2
Enter a number :: 3
Enter a number :: 4
Enter a number :: 5
Enter a number :: 6
Enter a number :: 7
Enter a number :: 8
Enter a number :: 9
Inserted matrix -->
1 2 3
4 5 6
7 8 9
Transpose of this given matrix is -->
1 4 7
2 5 8
3 6 9
```

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

```
#include <stdio.h>
int main(){
int n,i,j,sum=0;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n][n];

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
       printf("Enter a number :: ");
      scanf("%d",&arr[i][j]);

   }
}
printf("Inserted matrix -->\n");
for(i=0;i<n;i++){
   for(j=0;j<n;j++){
       printf("%d ",arr[i][j]);
      }
      printf("\n");
}

printf("sum of left diagonals ");
for(i=0;i<n;i++){
   j=i;
   printf("%d ",arr[i][j]);
}
printf("is :: ");

for(i=0;i<n;i++){
   j=i;
   sum += arr[i][j];
}
printf("%d",sum);
```

```
return 0;
}
```

Output::

```
enter size of array :: 3
Enter a number :: 1
Enter a number :: 2
Enter a number :: 3
Enter a number :: 4
Enter a number :: 5
Enter a number :: 6
Enter a number :: 7
Enter a number :: 8
Enter a number :: 9
Inserted matrix -->
1 2 3
4 5 6
7 8 9
sum of left diagonals 1 5 9 is :: 15
```

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

```
#include <stdio.h>
int main(){
int n,i,j,c=0,c2=0,res;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n][n];

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
      printf("Enter a number :: ");
     scanf("%d",&arr[i][j]);

   }
}

for(i=0;i<n;i++){
   j=i;
   if(arr[i][j] == 1){
      c +=1;
   }
}

if(c == n){
   for(i=0;i<n;i++){
      for(j=0;j<n;j++){
         if(arr[i][j] == 0){
           c2 += 1;
         }
      }
   }
}
```

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


res = (n * n) - n ;
if(c2 == res){
   printf("given matrix is identity matrix");
}
else{
  printf("given matrix is not a identity matrix");
}

return 0;
}
```

Output:

Case 1:

```
enter size of array :: 3
Enter a number :: 1
Enter a number :: 0
Enter a number :: 0
Enter a number :: 0
Enter a number :: 1
Enter a number :: 0
Enter a number :: 0
Enter a number :: 0
Enter a number :: 1
```

1 0 0
0 1 0
0 0 1
given matrix is identity matrix

case 2:

enter size of array :: 4
Enter a number :: 1
Enter a number :: 0
Enter a number :: 0
Enter a number :: 0
Enter a number :: 0
Enter a number :: 1
Enter a number :: 1
Enter a number :: 0
Enter a number :: 0
Enter a number :: 1
Enter a number :: 0
Enter a number :: 0
Enter a number :: 0
Enter a number :: 0
Enter a number :: 1
1 0 0 0
0 1 1 0
0 0 1 0
0 0 0 1
given matrix is not a identity matrix

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

```
#include<stdio.h>

int main(){
int i,n,j,k,f,search,temp=0;
printf("Enter size of a matrix :: ");
scanf("%d",&n);

int arr[n][n];

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
      printf("Enter a number :: ");
      scanf("%d",&arr[i][j]);
   }
}

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
     for(k=0;k<n;k++){
        for(f=0;f<n;f++){
           if(arr[i][j] < arr[k][f]){
              temp =arr[i][j];
              arr[i][j] = arr[k][f];
              arr[k][f] = temp;
           }

        }
     }
   }
```

```
}
printf("Which element do you want to search? :: ");
scanf("%d",&search);
printf("Sorted Matrix -->\n");
for(i=0;i<n;i++){
   for(j=0;j<n;j++){
      printf("%d ",arr[i][j]);
   }
   printf("\n");
}

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
      if(search == arr[i][j]){
         printf("%d is at index arr[%d][%d] in the sorted
matrix.",search,i,j);
      }
   }
   printf("\n");
}
   return 0;
}
```

Output:

```
Enter size of a matrix :: 3
Enter a number :: 5
Enter a number :: 6
Enter a number :: 4
Enter a number :: 9
Enter a number :: 8
Enter a number :: 7
Enter a number :: 3
```

Enter a number :: 1
Enter a number :: 2
Which element do you want to search? :: 4
Sorted Matrix -->
1 2 3
4 5 6
7 8 9

4 is at index arr[1][0] in the sorted matrix.