P1)
1- int size of the array
2- fill array with numbers → for Loop → scanf("%d" &num
3- get the number we need to find → scanf
4- Look for number in the array
5- if not found → y = -1 → printf("Not exist");
6- if X → found in the array → Brake + (y+1)
7- if y != -1 → printf("yes \n", y)
8- if y = -1 → printf("Not exist")

P2)
1- get the Size of Array → int n + &scanf("%d", n)
2- Take array Numbers → For Loop + Scanf
3- Make int of smallest num and it's possition
4- For(i=1 ; i<n ; i++) { if (smallest > numbers[i])
{ smallest = numbers[i];
position = i+1
5- printf ("%d %d", smallest, position);

P3)
1- Read input Array
2- Sorting: Sort array in ascending order, Comparing adjacent & swap them if they are in wrong order

```
if (arr[J] > arr[J+1] {
    int temp= arr[J]
    arr[J] = arr[J+1]
    arr[J+1] = temp;
```

3- print array after sorting

p4) A: reading Both matrixes

2. Adding first mat to second

```
For( int i=0 ; i<3 ; i++ )
for ( int J=0 ; J<3 ; J++ ) } 1
    Sum[i][J] = matrix 1[i][J] + matrix 2[i][J]
```

3. print results: For (1 again) printf("%d", Sum[i][J]);

p5) 1. Take the matrix Size from User in (n)

2. adding the matrix elements

3. Calculating the sum of main diagonal.

main diagonal += matrix[i][i];

4. " " " " Secondary diagonal

Secondary diagonal += matrix[i][n-i-1]

5. Calculating result: int result = abs (main dig - Sec dig);

6. print the result.

p6) mirror array: 1. input matrix element

2. print reversed output.

```
For( int i=0 ; i<n ; i++)
For(int J=m-1 ; J>=0 ; J--)
printf("%d", matrix[i][J]);
```