AGNIM GUPTA
2028083
CSSE
A23
Q1

```c
#include <stdio.h>

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

void sort(int arr[], int size)
{
    int left = 0, right = size-1;
    while (left < right)
    {
        while (arr[left]%2 == 0 && left < right)
            left++;

        while (arr[right]%2 == 1 && left < right)
            right--;

        if (left < right)
        {
            swap(&arr[left], &arr[right]);
            left++;
            right--;
        }
    }
}



int main()
{
    int arr[10];
    printf("enter an array of size 10: ");
    for(int j=0; j<10; j++)
    {
        scanf("%d", &arr[j]);
    }
    int arr_size = sizeof(arr)/sizeof(arr[0]);
    int i = 0;

    sort(arr, arr_size);

    printf("Array after segregation ");
    for (i = 0; i < arr_size; i++)
        printf("%d ", arr[i]);

    return 0;
}
```

```
PS C:\Users\KIIT\Documents\coding> cd "c:\Users\KI
IT\Documents\coding\3rd semister\DSA lab\class 2\"
 ; if ($?) { gcc class2_q1.c -o class2_q1 } ; if (
$?) { .\class2_q1 }
enter an array of size 10: 3
543
64
23
98
32
64
15
20
81
Array after segregation 20 64 64 32 98 23 543 15 3
 81
PS C:\Users\KIIT\Documents\coding\3rd semister\DSA
 lab\class 2>
```

Q2

```c
#include <stdio.h>

int main()
{
    int n;
    printf("Enter size of square matrix: ");
    scanf("%d", &n);
    printf("Enter %d elements of square matrix: ", n*n);
    int arr[n][n];
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<n; j++)
        {
            int p;
            scanf("%d", &p);
            arr[i][j]=p;
        }
    }
    int cnt=0;
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<n; j++)
        {
            if(arr[i][j]!=0)
            {
                cnt++;
            }
        }
    }
    printf("Number of non zero elements in array are %d \n", cnt);
    int sum=0;
    for(int i=0; i<n; i++)
    {
        sum += arr[i][i];
    }
    printf("Sum of leading diagonal elements are %d \n", sum);
    int temp=n-1;
    printf("Minor diagonal elements are: ");
    for(int i=0; i<n; i++)
    {
        int p;
        p=arr[i][temp];
        printf("%d, ", p);
        temp--;
    }
    int product=1;
    for(int i=0; i<n; i++)
    {
        product *= arr[i][i];
    }
    printf("\nProduct of diagonal elements are: %d \n", product);
}
```

```
PS C:\Users\KIIT\Documents\coding\3rd semister\DSA lab
\class 2> cd "c:\Users\KIIT\Documents\coding\3rd semis
ter\DSA lab\class 2\" ; if ($?) { gcc class2_q2.c -o c
lass2_q2 } ; if ($?) { .\class2_q2 }
Enter size of square matrix: 2
Enter 4 elements of square matrix: 1
2
3
4
Number of non zero elements in array are 4
Sum of leading diagonal elements are 5
Minor diagonal elements are: 2, 3,
Product of diagonal elements are 4
PS C:\Users\KIIT\Documents\coding\3rd semister\DSA lab
\class 2>
```

Q3

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

// Function to return k'th smallest element in a given array
int kthSmallest(int arr[], int n)
{
    int c, d, t;

    for (c = 0 ; c < ( n - 1 ); c++)
    {
        for (d = 0 ; d < n - c - 1; d++)
        {
            if (arr[d] > arr[d+1])
            {
                /* Swapping */

                t         = arr[d];
                arr[d]    = arr[d+1];
                arr[d+1] = t;
            }
        }
    }
}

// Driver program to test above methods
int main()
{
    int arr[] = { 12, 3, 5, 7, 19 };
    int n = sizeof(arr) / sizeof(arr[0]);
    kthSmallest(arr, n);
    printf("K'th smallest element is %d\n", arr[0]);
    printf("K'th largest element is %d", arr[n-1]);
    return 0;
}
```

```
PS C:\Users\KIIT\Documents\coding> cd "c:\Users\KIIT\Documents\coding\3rd semister\DSA lab\class 2\" ; if ($?) { gcc class2_q3.c -o class2_q3 } ; if ($?) { .\class2_q3 }
K'th smallest element is 3
K'th largest element is 19
PS C:\Users\KIIT\Documents\coding\3rd semister\DSA lab\class 2>
```

Q4

```c
#include <stdio.h>
#define n 4

void change(int arr[][n])
{
    for (int i = 0; i < n; i++)
    {
        for(int j=0; j<n/2; j++)
        {
            int t = arr[i][j];
            arr[i][j] = arr[i][n - j-1];
            arr[i][n-j-1] = t;
        }
    }
}

int main()
{
    int arr[n][n] = { { 34, 58, 21, 54 },
                      { 47, 97, 34, 25 },
                      { 35, 22, 14, 86 },
                      { 15, 23, 43, 71 } };

    change(arr);

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

```
PS C:\Users\KIIT\Documents\coding> cd "c:\Users\KIIT\D
ocuments\coding\3rd semister\DSA lab\class 2\" ; if ($
?) { gcc class2_q4.c -o class2_q4 } ; if ($?) { .\clas
s2_q4 }
54 21 58 34
25 34 97 47
86 14 22 35
71 43 23 15
PS C:\Users\KIIT\Documents\coding\3rd semister\DSA lab
\class 2>
```