1. Write a function to take a floating point number as input and returns the same number rounded to k decimal places. Do not use any system defined functions.
   For example: If n=17.24578 and k = 2, the output is 17.25
   If n= 345.2034 and k=3 then output is 345.203

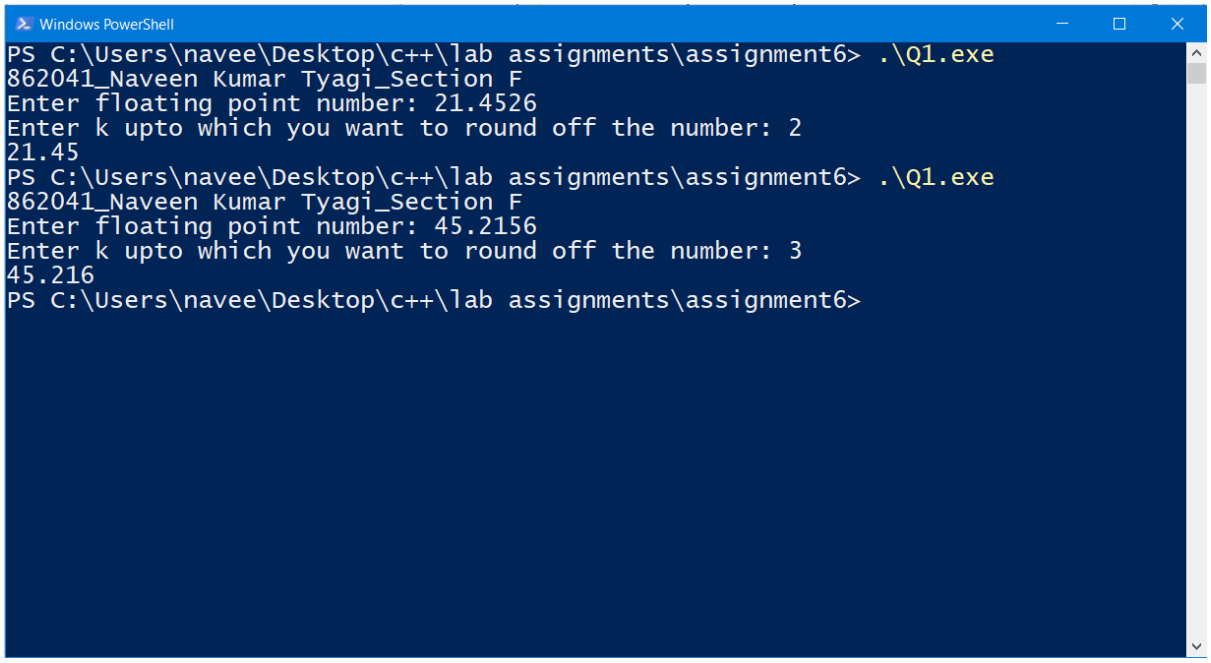
Code:

```
//862041_Naveen Kumar Tyagi_Section F

#include<iostream>

using namespace std;



//funtion to raise the power of 10 to k

float pow_10(int k){

   if(k==0){

      return 1;

   }

   return 10*pow_10(k-1);

}



//funtion which round off the number

//upto desired decimal places

//first multiply the number by

//10 raised to the power k

//then extract decimal part of the new number(intermediate variable)

//if decimal part>0.5 then increase integral part of new num by 1

//if it is less than 0.5 then left as it is
```

```
//after that divide the integral part by
//10 raised to power k
//then we get desired reult
float round_off(float num,int k){
   float itrmd=num*pow_10(k);
   int int_itrmd=static_cast<int>(itrmd); //storing integral part of intermediate var
   float decimal_part=itrmd-int_itrmd;    //storing decimal part of intermediat var
   if(decimal_part>=0.5){
      int_itrmd+=1;
   }
   num=int_itrmd/pow_10(k);
   return num;
}

int main(){
   cout<<"862041_Naveen Kumar Tyagi_Section F";
   float num;
   int k;
   cout<<"\nEnter floating point number: ";
   cin>>num;
   cout<<"Enter k upto which you want to round off the number: ";
   cin>>k;
   cout<<round_off(num,k);
```

```
    return 0;

}
```

Output:



2. Write a menu-driven program that allows a user to enter five numbers and then choose between finding the smallest, largest, sum, or average. The menu and all the choices are to be functions. Use a switch statement to determine what action to take. Provide an error message if an invalid choice is entered.

Code:

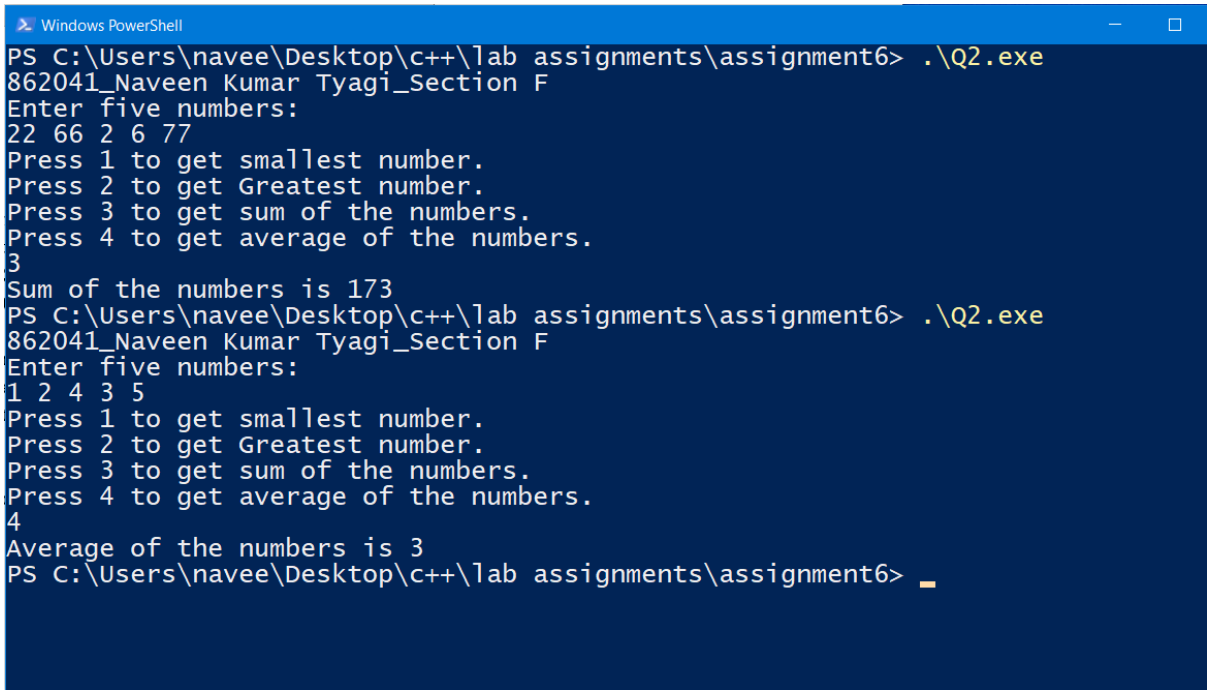
```
//862041_Naveen Kumar Tyagi_Section F

#include<iostream>

using namespace std;



//function to find the smallest number in the array

int smallest(int arr[],int n){

   int smallest=arr[0];   //assuming first element be smallest

   for(int i=1;i<n;i++){      // loop to find and store the smallest

      if(smallest>arr[i]){

         smallest=arr[i];

      }

   }

   return smallest;

}



//function to find the greatest number in the array

int greatest(int arr[],int n){

   int greatest=arr[0];  //assuming first element be greatest

   for(int i=1;i<n;i++){    // loop to find and store greatest
```

```
        if(greatest<arr[i]){
            greatest=arr[i];
        }
    }
    return greatest;
}

//function to get sum of array
int sum(int arr[],int n){
    int sum=0;
    for(int i=0;i<n;i++){
        sum+=arr[i];  //add element and sum of previous element and store in sum
    }
    return sum;
}
//function to find average
float average(int arr[],int n){
    int int_sum=sum(arr,n);  //storing sum of elements
    float sum=static_cast<float>(int_sum);  //converting datatype of sum to float
    float average=sum/n;   //storing average;
    return average;
}
```

```
//menu function
void menu(int arr[],int n){
   int response;  //to store input of user in integers
   cout<<"Press 1 to get smallest number.\n"
   <<"Press 2 to get Greatest number.\n"
   <<"Press 3 to get sum of the numbers.\n"
   <<"Press 4 to get average of the numbers.\n";
   cin>>response;

   //switch statement to pass the entered numbers
   //to required funtion to get desired result
   switch(response){
      case 1:
      cout<<"Smallest number is "<<smallest(arr,n);
      break;    //break the statement if above case found as no need to others then
      case 2:
      cout<<"Greatest numbers is "<<greatest(arr,n);
      break;
      case 3:
      cout<<"Sum of the numbers is "<<sum(arr,n);
      break;
      case 4:
      cout<<"Average of the numbers is "<<average(arr,n);
```

```
        break;
        default:  //default case for invalid input is given
        cout<<"Entered choice is invalid. Enter correct one.";
        menu(arr,n);
    }
}
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    int arr[5];
    cout<<"Enter five numbers: \n";
    for(int i=0;i<5;i++){   //storing numbers in array
        cin>>arr[i];
    }
    menu(arr,5);  //call of menu function
    return 0;
}
```

Output:


Windows PowerShell
PS C:\Users\navee\Desktop\c++\lab assignments\assignment6> .\Q2.exe
862041_Naveen Kumar Tyagi_Section F
Enter five numbers:
22 66 2 6 77
Press 1 to get smallest number.
Press 2 to get Greatest number.
Press 3 to get sum of the numbers.
Press 4 to get average of the numbers.
3
Sum of the numbers is 173
PS C:\Users\navee\Desktop\c++\lab assignments\assignment6> .\Q2.exe
862041_Naveen Kumar Tyagi_Section F
Enter five numbers:
1 2 4 3 5
Press 1 to get smallest number.
Press 2 to get Greatest number.
Press 3 to get sum of the numbers.
Press 4 to get average of the numbers.
4
Average of the numbers is 3
PS C:\Users\navee\Desktop\c++\lab assignments\assignment6>

3. Write a C++ program to find common elements from three sorted (in non-decreasing order) arrays.
   For example:
   array1 = 2, 4, 8
   array2 = 2, 3, 4, 8, 10, 16
   array3 = 4, 8, 14, 40
   Output:
   Common elements from three sorted (in non-decreasing order) arrays: [4, 8]

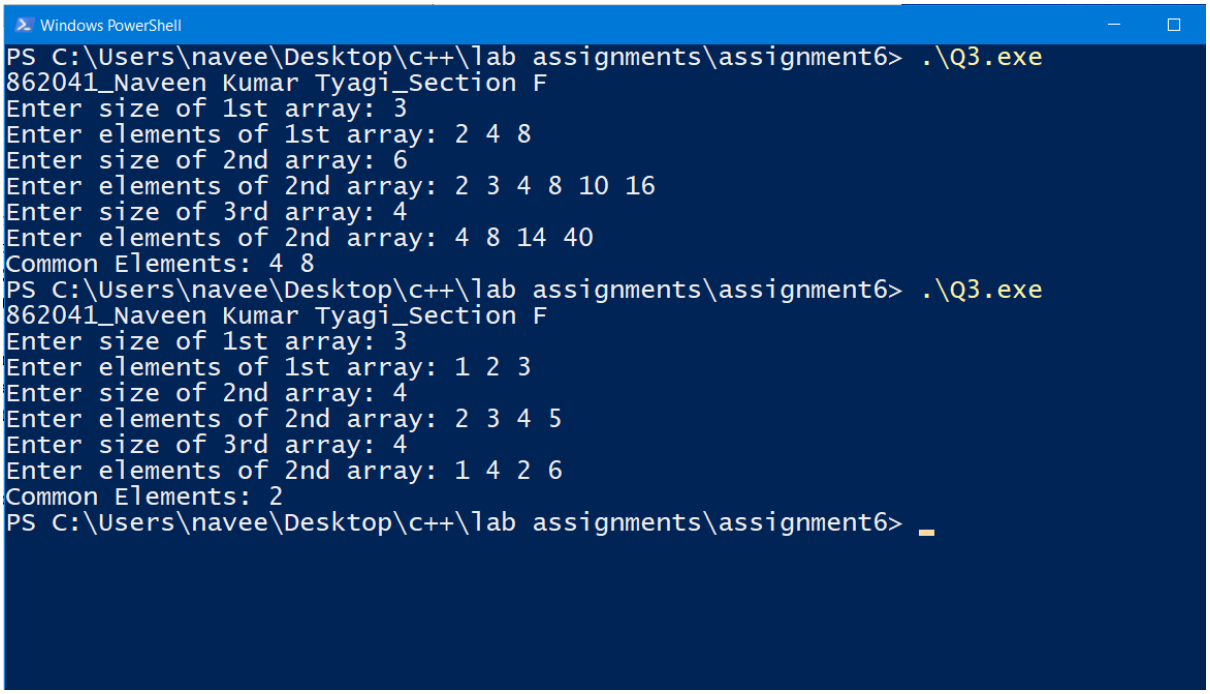
Code:

```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
using namespace std;

//comm array is to store common elements
//counter variable is to store no. of common variables
int comm[10],counter=0;

//function to find common elements and store in comm array
//it pick an element from 1st array
//and search in 2nd array
//if found in 2nd then it will search in 3rd array
//if found in 3rd array then it will store
//if not found in 2nd array then there is need to search in 3rd array
//then it start searching 2nd element of 1st array in same procedure
void comm_elem_finder(int arr1[], int arr2[], int arr3[],int a1,int a2, int a3){
   for(int i=0;i<a1;i++){      //pick elements from 1st array
      for(int j=0;j<a2;j++){    //search in 2nd array
         if(arr1[i]==arr2[j]){   // if found
            for(int k=0;k<a3;k++){  //search in 3rd
```

```
                if(arr1[i]==arr3[k]){  //if found then store
                    comm[counter]=arr1[i];
                    counter++;
                }
            }
        }
    }
}
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    int a1,a2,a3;

    cout<<"Enter size of 1st array: ";
    cin>>a1;
    int arr1[a1];    //1st array declartion
    cout<<"Enter elements of 1st array: ";
    for(int i=0;i<a1;i++){   //taking input for 1st array
        cin>>arr1[i];
    }

    cout<<"Enter size of 2nd array: ";
    cin>>a2;
    int arr2[a2];         //2nd array declaration
    cout<<"Enter elements of 2nd array: ";
    for(int i=0;i<a2;i++){   //taking input for 2nd array
        cin>>arr2[i];
    }
    cout<<"Enter size of 3rd array: ";
```

```
    cin>>a3;
    int arr3[a3];      //3rd array declaration
    cout<<"Enter elements of 2nd array: ";
    for(int i=0;i<a3;i++){    //taking input for 3rd array
        cin>>arr3[i];
    }
    cout<<"Common Elements: ";
    //passing input to function that give common elements
    comm_elem_finder(arr1,arr2,arr3,a1,a2,a3);

    //prints the common elements
    for(int i=0;i<counter;i++){
        cout<<comm[i]<<" ";
    }
    return 0;
}
```

Output:



4. Suppose A, B, C are arrays of integers of size M, N, and M + N respectively. The numbers in array A appear in ascending order while the numbers in array B appear in descending order. Write a user defined function in C++ to produce third array C by merging arrays A and B in ascending order. Use A, B and C as arguments in the function.

Output:

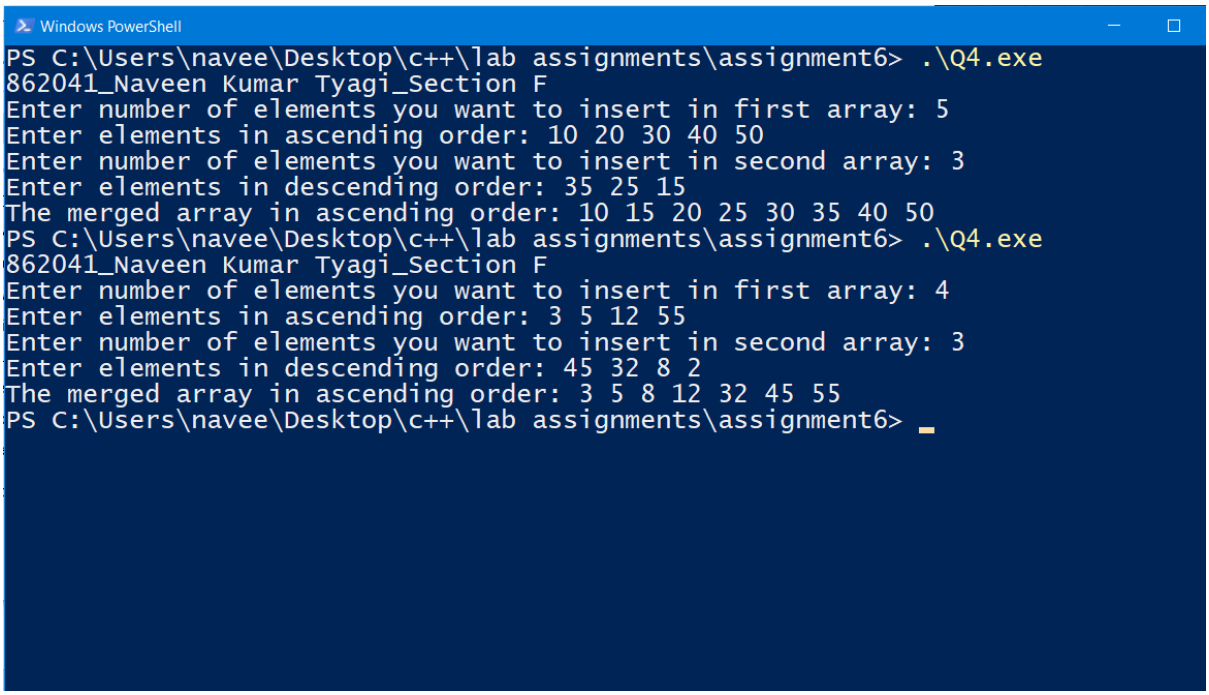
Enter number of elements you want to insert in first array 5
Enter element in ascending order
10 26 30 44 45
Enter number of elements you want to insert in second array 3
Enter element in descending order
44 40 26
The Merged Array in ascending Order
10 26 30 34 40 44 45

Code:

```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
using namespace std;

//function to merge two array and store in third one
void array_merger(int A[],int B[],int C[],int M,int N){
   for(int i=0;i<M;i++){  //store 1st array in 3rd
      C[i]=A[i];
   }
   for(int i=0;i<N;i++){   //store 2nd array in 3rd
      C[M+i]=B[i];
   }
   //selection sort to sort the third array
   for(int i=0;i<M+N-1;i++){
      int loc=i;
      int min=C[i];  //assuming first element of unsorted subarray minimum
      for(int j=i;j<M+N;j++){  //finding min and storing it's location
         if(min>C[j]){
            loc=j;
            min=C[j];
         }
      }
      //swapping min with first unsorted subarray element
      int temp=C[i];
      C[i]=C[loc];
```

```
        C[loc]=temp;
    }
}
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    int M,N;
    cout<<"Enter number of elements you want to insert in first array: ";
    cin>>M;
    int A[M]; //declaration of 1st array
    cout<<"Enter elements in ascending order: ";
    for(int i=0;i<M;i++){  //taking input for 1st array
        cin>>A[i];
    }
    cout<<"Enter number of elements you want to insert in second array: ";
    cin>>N;
    int B[N];  //declaration of 2nd array
    cout<<"Enter elements in descending order: ";
    for(int i=0;i<N;i++){   //taking input for 2nd array
        cin>>B[i];
    }
    int C[M+N];   //declaration of third array
    array_merger(A,B,C,M,N);   //passing arrays to merge them
    cout<<"The merged array in ascending order: ";
    //prints the merged array (third)
    for(int i=0;i<M+N;i++){
        cout<<C[i]<<" ";
    }
    return 0;
}
```

Output:


Windows PowerShell
PS C:\Users\navee\Desktop\c++\lab assignments\assignment6> .\Q4.exe
862041_Naveen Kumar Tyagi_Section F
Enter number of elements you want to insert in first array: 5
Enter elements in ascending order: 10 20 30 40 50
Enter number of elements you want to insert in second array: 3
Enter elements in descending order: 35 25 15
The merged array in ascending order: 10 15 20 25 30 35 40 50
PS C:\Users\navee\Desktop\c++\lab assignments\assignment6> .\Q4.exe
862041_Naveen Kumar Tyagi_Section F
Enter number of elements you want to insert in first array: 4
Enter elements in ascending order: 3 5 12 55
Enter number of elements you want to insert in second array: 3
Enter elements in descending order: 45 32 8 2
The merged array in ascending order: 3 5 8 12 32 45 55
PS C:\Users\navee\Desktop\c++\lab assignments\assignment6>

5. Write a program to print the elements of a 2d array in the form of a matrix in spiral form.
   For Example:
   Input: 1 2 3 4
   5 6 7 8
   9 10 11 12
   13 14 15 16
   Output: 1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

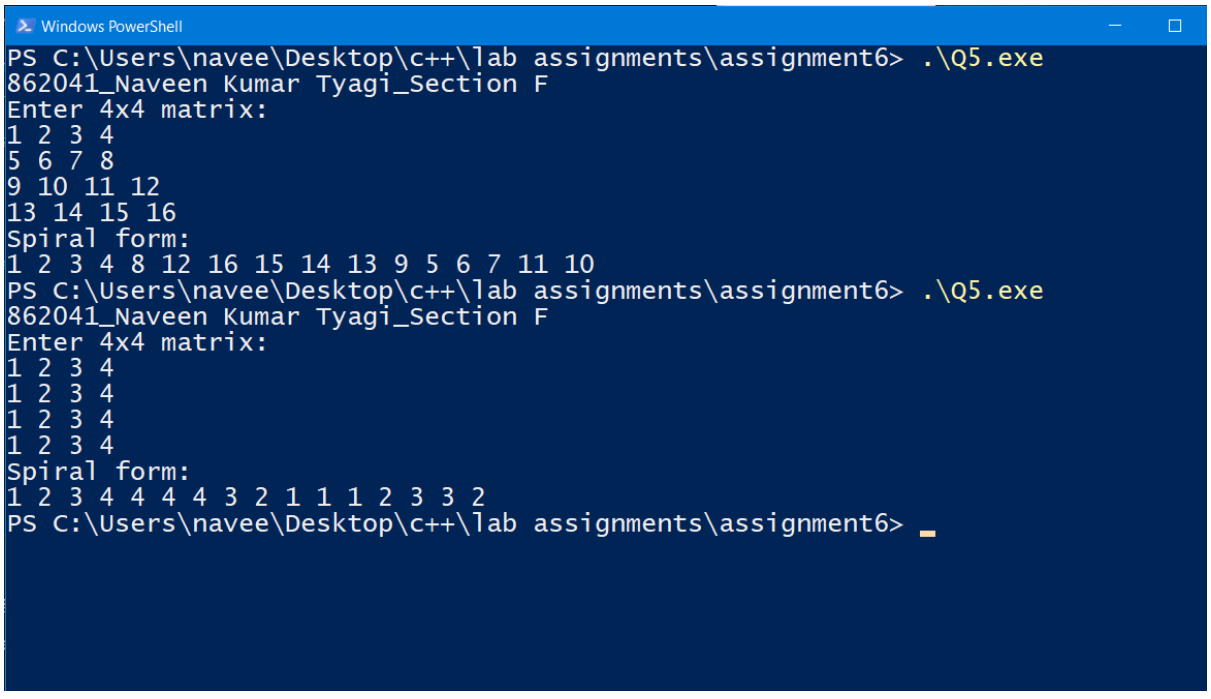
Code:

```
//862041_Naveen Kumar Tyagi_Section F

#include<iostream>

using namespace std;



//function to print matrix in spiral form

//r_s is first row index

//c_s is first column index

//r_e is last row index

//s_e is last column index

void spiral_printer(int arr[4][4],int r_s,int c_s,int r_e,int c_e){

   if(r_s<=r_e &&c_s<=c_e){

      //print first row

      for(int j=c_s;j<=c_e;j++){

         cout<<arr[r_s][j]<<" ";

      }

      r_s++;

      //print last column
```

```
        for(int i=r_s;i<=r_e;i++){
            cout<<arr[i][c_e]<<" ";
        }
        c_e--;
        //print last row if it not equal to first row
        if((r_s-1)!=r_e){
          for(int j=c_e;j>=c_s;j--){
              cout<<arr[r_e][j]<<" ";
          }
          r_e--;
        }
        //print first column if it is equal to first
        if(c_s!=(c_e+1)){
          for(int i=r_e;i>=r_s;i--){
              cout<<arr[i][c_s]<<" ";
          }
          c_s++;
        }
        spiral_printer(arr,r_s,c_s,r_e,c_e); //call itself to print left part
    }
}
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
```

```
    int arr[4][4];/*={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};*/
    cout<<"Enter 4x4 matrix:\n";
    for(int i=0;i<4;i++){ //take input and store in array
        for(int j=0;j<4;j++){
            cin>>arr[i][j];
        }
    }
    //call function to print matrix in spiral form
    cout<<"Spiral form:\n";
    spiral_printer(arr,0,0,3,3);
    return 0;
}
```

Output:



6. Write a user defined function named upper_half() which takes a two dimensional array A, with size N rows and N columns as argument and prints the upper half of the array.

   For e.g.,

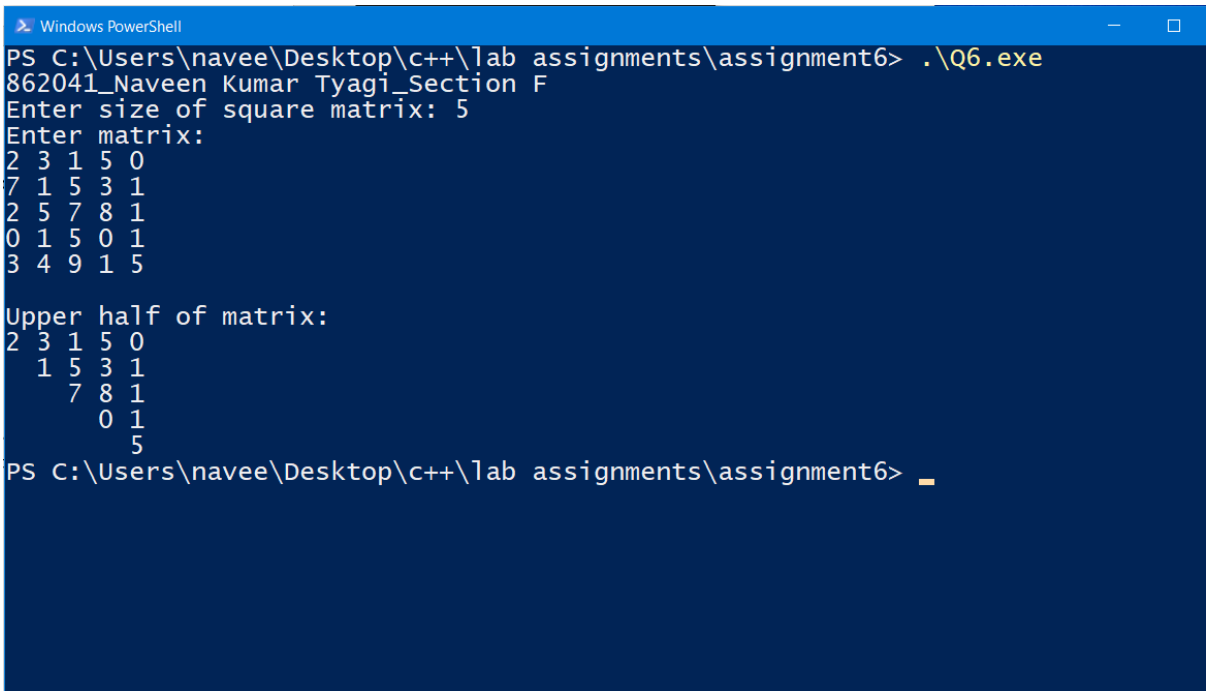
```
2 3 1 5 0                    2 3 1 5 0
7 1 5 3 1                      1 5 3 1
2 5 7 8 1   Output will be:    1 7 8
0 1 5 0 1                        0 1
3 4 9 1 5                          5
```

Code:

```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
using namespace std;

//function to print upper half
void upper_half(int n,int mat[10][10]){
   for(int i=0;i<n;i++){
      for(int j=0;j<i;j++){ //print space
         cout<<"  ";
      }
      for(int j=i;j<n;j++){  // print row elements starting from that element
         cout<<mat[i][j]<<" "; //of which column number is equal to row number
      }
      cout<<endl;
   }
}
int main(){
   cout<<"862041_Naveen Kumar Tyagi_Section F\n";
   int n;
   cout<<"Enter size of square matrix: ";
```

```
    cin>>n;   //store size of square matrix
    int mat[10][10];  // declaration if 2d array to store elements of matrix
    cout<<"Enter matrix:\n";
    for(int i=0;i<n;i++){  //take inputs for matrix and store in 2d array
        for(int j=0;j<n;j++){
            cin>>mat[i][j];
        }
    }
    cout<<"\nUpper half of matrix:\n";
    //call function that print upper half of the matrix
    upper_half(n,mat);
    return 0;
}
```

Output:



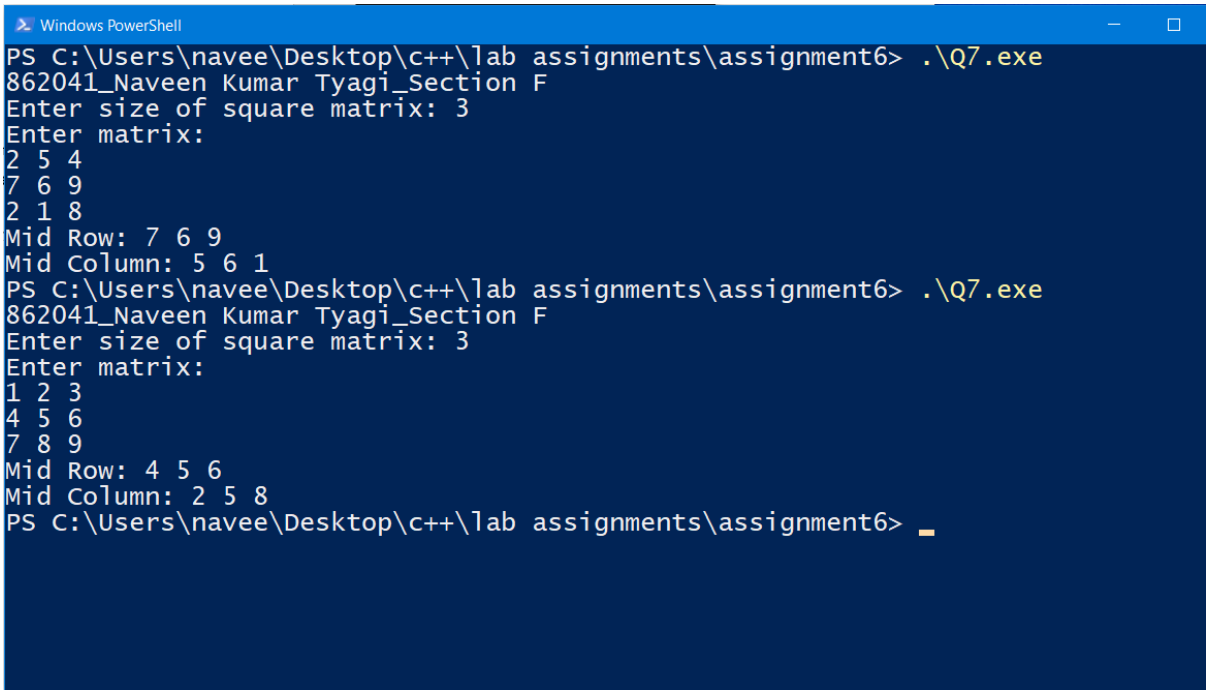
7. Write a function in C++ which accepts a 2D array of integers and its size as arguments and displays the elements of middle row and the elements of middle column.
   [Assuming the 2D Array to be a square matrix with odd dimension i.e. 3x3, 5x5, 7x7 etc...]
   Example, if the array contents are
   3 5 4
   7 6 9
   2 1 8
   Output through the function should be:
   Middle Row : 7 6 9
   Middle column : 5 6 1

Code:

```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
using namespace std;

//function to print mid row and column
void mid_row_column(int mat[10][10],int n){
   int mid_row_no=n/2;  //index of mid row
   int mid_column_no=n/2; //index of mid column
   cout<<"Mid Row: ";

   //print mid row
   for(int j=0;j<n;j++){
      cout<<mat[mid_row_no][j]<<" ";
   }
   cout<<endl;
   cout<<"Mid Column: ";

   //print mid column
   for(int i=0;i<n;i++){
```

```
        cout<<mat[i][mid_column_no]<<" ";
    }
}
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    int n;
    cout<<"Enter size of square matrix: ";
    cin>>n;
    int mat[10][10]; //declaration of 2d array to store matrix
    cout<<"Enter matrix:\n";
    for(int i=0;i<n;i++){      //take input and store in 2d array
        for(int j=0;j<n;j++){
            cin>>mat[i][j];
        }
    }
    //call function to print mid row and column of matrix
    mid_row_column(mat,n);
    return 0;
}
```

Output:


Windows PowerShell
PS C:\Users\navee\Desktop\c++\lab assignments\assignment6> .\Q7.exe
862041_Naveen Kumar Tyagi_Section F
Enter size of square matrix: 3
Enter matrix:
2 5 4
7 6 9
2 1 8
Mid Row: 7 6 9
Mid Column: 5 6 1
PS C:\Users\navee\Desktop\c++\lab assignments\assignment6> .\Q7.exe
862041_Naveen Kumar Tyagi_Section F
Enter size of square matrix: 3
Enter matrix:
1 2 3
4 5 6
7 8 9
Mid Row: 4 5 6
Mid Column: 2 5 8
PS C:\Users\navee\Desktop\c++\lab assignments\assignment6>