Logic Building Assignment : 10

Complete below code snippets it contains only service provider function.

Write entry point function to call below helper functions separately.

Create separate visual Studio project for each problem statement separately.

Each project should contains below things
- **File which contains entry point function**
- **File which contains helper function**
- **File which works as header file**

1. Accept two arrays from user and display contents of each array.

```
Input :   2    9    7    5    2    3
          9    3    5    5

Output :  2    9    7    5    2    3
          9    3    5    5
```

```
void Display(int arr1[], int iSize1, int arr2[], int iSize2)
{
        // Traverse Both the array and display its contents
}
```

2. Accept two arrays from user and display even contents of each array.

```
Input :   2    9    6    5    2    3
          45   6    12   18   23   4

Output :  2    6    2
          6    12   18   4
```

```
void DisplayEven(int arr1[], int iSize1, int arr2[], int iSize2)
{
        // Traverse Both the array and display its even contents
}
```

4. Accept two arrays from user and form new array which is combination of contents of first and second array.

Input : 12 57 28 3
99 23 54 58 6 67

Output : 12 57 28 3 99 23 54 58 6 67

```
int * ArrayConcate(int arr1[], int iSize1, int arr2[], int iSize2)
{
	// Allocate dynamic memory to store contents of both arrays
	// Traverse Both the array and copy contents into new array
	// Return address of new array
}
```

5. Accept two arrays from user and display summation of each array.

Input : 2 9 7 5 2 3
9 3 5 5

Output : 28 22

```
void SumArray(int arr1[], int iSize1, int arr2[], int iSize2)
{
	// Traverse Both the array and display its summation
}
```

6. Accept two arrays from user and return difference between summation of arrays.

Input : 2 9 7 5 2 3
9 3 5 5

Output : 6

```
int DiffArray(int arr1[], int iSize1, int arr2[], int iSize2)
{
	// Traverse Both the array and calculate its summation
	// Return difference between summation
}
```

7. Accept two arrays from user and display minimum element of each array.

Input : 2 9 7 5 2 3
9 3 5 5

Output : 2 3

```
void MinArray(int arr1[], int iSize1, int arr2[], int iSize2)
{
	// Traverse Both the array and display its summation
}
```

8. Accept array from user and copy the contents of that array into another array in reverse order and return base address of new array.

```
int * CopyArrayRev (int arr[], int iSize)
{
	// Allocate memory for new array using malloc

	// Copy the contents from input array into new array in reverse order

	// Return base address of dynamic memory
}
```

9. Accept array from user and copy the contents of that array into another array and return base address of new array.

```
int * CopyArray (int arr[], int iSize)
{
	// Allocate memory for new array using malloc

	// Copy the contents from input array into new array

	// Return base address of dynamic memory
}
```

10. Accept array from user and check whether that array is palindrome or not.

Input : 10 25 38 25 10

Output : TRUE

Input : 10 25 38 78 25 10

Output : FALSE

```
BOOL ChkPallindrome (int arr[], int iSize)
{
                // Logic
}
```