

TASK 1: Create an array A of length 10 integers. Values ranging from 1 to 50.

- 1. Find all pair of elements whose sum is 25.**
 - 2. Find the number of elements of A which are even, and the number of elements of A which are odd.**
 - 3. Write a procedure which finds the average of the value of A.**
-

SOLUTION :

```
static void Main(string[] args)
{
    int[] a = { 12, 1, 15, 10, 35, 7, 22, 3, 9, 41 };
    int even = 0, odd = 0, total = 0;
    double avg;

    // PAIRS WITH SUM OF 25
    for (int i = 0; i < a.Length; i++)
    {
        for (int j = i; j < a.Length; j++)
        {
            if (a[i] + a[j] == 25)
            {
                Console.WriteLine("The values at {0} and {1} indexes that are {2} and {3} make up a sum of 25", i, j, a[i], a[j]);
            }
        }
    }

    Console.WriteLine("\n\n");
```

```
//ELEMENTS OF A WHICH ARE EVEN AND ODD
```

```
for (int i = 0; i < a.Length; i++)  
{  
    if (a[i] %2 == 0)  
    {  
        Console.WriteLine("{0} at index {1} is even", a[i], i);  
        even++;  
    }  
    else  
    {  
        Console.WriteLine("{0} at index {1} is odd", a[i], i);  
        odd++;  
    }  
}
```

```
Console.WriteLine("\n {0} numbers from the array of 10 integers are even \n{1} numbers from the  
array of 10 integers are odd", even, odd);
```

```
Console.WriteLine("\n\n");
```

```
//AVERAGE OF THE VALUES OF A
```

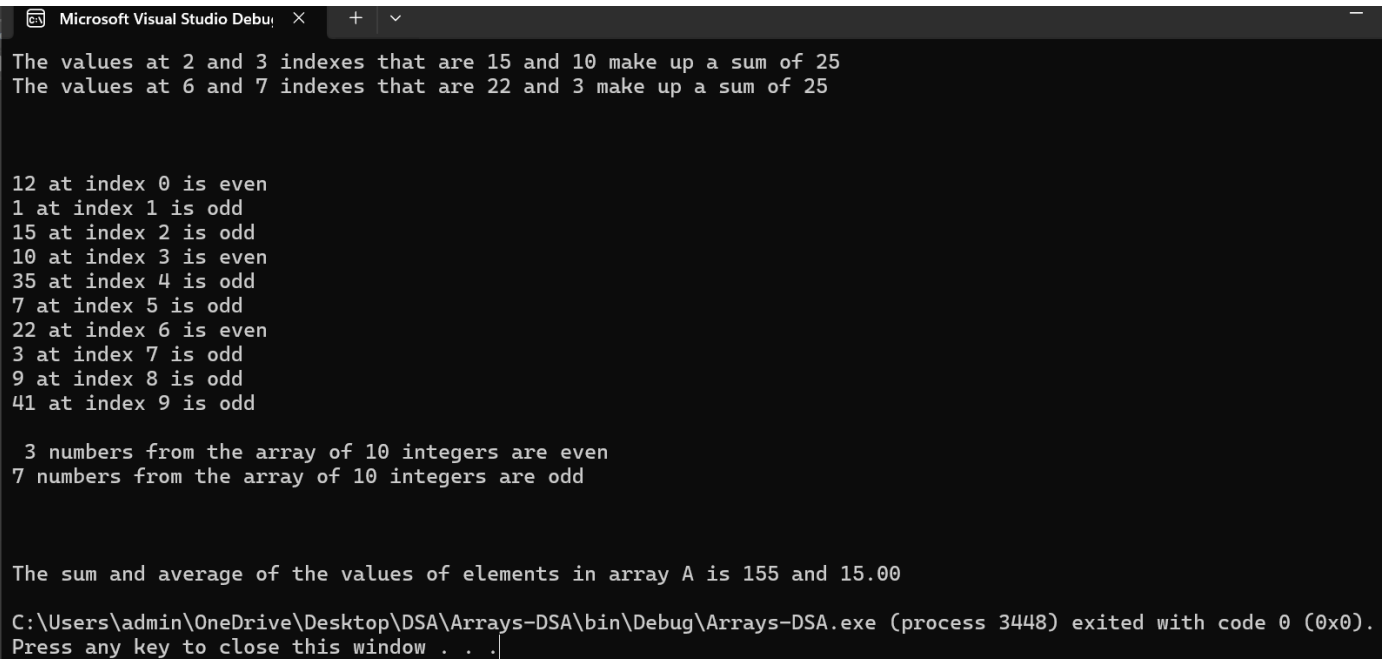
```
for(int i = 0; i<a.Length; i++)
```

```
{  
    total += a[i];  
}
```

```
avg = total / 10;
```

```
Console.WriteLine("The sum and average of the values of elements in array A is {0} and {1:F2}",  
total, avg);
```

```
}
```

OUTPUT :The image shows a screenshot of the Microsoft Visual Studio Debug Console window. The window has a dark background and a title bar that reads "Microsoft Visual Studio Debug Console". The output text is as follows:

```
The values at 2 and 3 indexes that are 15 and 10 make up a sum of 25
The values at 6 and 7 indexes that are 22 and 3 make up a sum of 25

12 at index 0 is even
1 at index 1 is odd
15 at index 2 is odd
10 at index 3 is even
35 at index 4 is odd
7 at index 5 is odd
22 at index 6 is even
3 at index 7 is odd
9 at index 8 is odd
41 at index 9 is odd

3 numbers from the array of 10 integers are even
7 numbers from the array of 10 integers are odd

The sum and average of the values of elements in array A is 155 and 15.00

C:\Users\admin\OneDrive\Desktop\DSA\Arrays-DSA\bin\Debug\Arrays-DSA.exe (process 3448) exited with code 0 (0x0).
Press any key to close this window . . .
```

TASK 2: Write a C# program that utilizes a 1D array to implement a simple inventory management system

1. Inventory Setup

Create an array to store inventory items and initialize the item count.

2. Main Menu Loop

Implement the main menu loop for the inventory management system. This loop will repeatedly display options to the user until they choose to exit.

3. Add Item

Implement the functionality to add an item to the inventory. Ask the user for the name of the item to add and add it to the inventory array.

4. Remove Item

Implement the functionality to remove an item from the inventory. Ask the user for the name of the item to remove and remove it from the inventory array if found.

5. Search Item

Implement the functionality to search for an item in the inventory. Ask the user for the name of the item to search for and display whether it's in the inventory or not.

6. Display Inventory

Implement the functionality to display the current items in the inventory.

7. Exit Program

Implement the functionality to exit the inventory management system when the user chooses to exit.

SOLUTION :

```
static void Main(string[] args)
{
    Console.Write("Specify the size of the inventory: ");
    int size = Convert.ToInt32(Console.ReadLine());

    String[] inventory = new String[size];
    int count = 0;
    String itemName;
    char exit;

    Console.Write("Enter the number of inventory items you want to setup? : ");
    int items = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("\n");
    if (items < size)
    {
        for (int i = 0; i < items; i++)
        {
            Console.Write("Enter the item name: ");
            inventory[count] = Console.ReadLine();
            count++;
        }
        Console.WriteLine("Successful Inventory Setup");
    }
    else
    {
        Console.WriteLine("Items limit is exceeded");
    }
}
```

```
do
{
    Console.WriteLine("\n");

    Console.Write("Action list: \n 1.Add Items \n 2.Remove Items \n 3.Search Items \n 4.Display
Inventory \nSelect actions: ");

    int action = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("\n");
    switch (action)
    {
        case 1:
            Console.Write("Enter the number of items you want to add in the inventory: ");
            items = Convert.ToInt32(Console.ReadLine());
            if ((count + items) <= size)
            {
                for (int i = 0; i < items; i++)
                {
                    Console.Write("Enter the item name: ");
                    inventory[count] = Console.ReadLine();
                    count++;
                }
                Console.WriteLine("Items added successfully");
            }
            else
            {
                Console.WriteLine("Items exceeds inventory limit");
            }
            break;
```

case 2:

```
Console.Write("Enter the number of items you want to remove from the inventory: ");
items = Convert.ToInt32(Console.ReadLine());
bool itemsRemoved = false;
if (items < count)
{
    if ((count - items) >= 0)
    {
        for (int i = 0; i < items; i++)
        {
            Console.Write("Enter the item name: ");
            itemName = Console.ReadLine();
            for (int j = 0; j < count; j++)
            {
                if (itemName == inventory[j])
                {
                    for (int k = j; k < count - 1; k++)
                    {
                        inventory[k] = inventory[k + 1];
                    }
                    --count;
                    itemsRemoved = true;
                    Console.WriteLine("Item removed successfully");
                }
            }
        }
    }
}
```

```
        if (itemsRemoved != true)
        {
            Console.WriteLine("Selected item not found in the inventory");
        }
    }
}
else
{
    Console.WriteLine("Items exceeds minimum inventory limit");
}
}
else
{
    Console.WriteLine("Invenotry does not contain {0} items", items);
}
break;
```

case 3:

```
    Console.Write("Enter the item name: ");
    itemName = Console.ReadLine();
    bool itemFound = false;
    for (int i = 0; i < count; i++)
    {
        if (itemName == inventory[i])
        {
            itemFound = true;
            Console.WriteLine("Item found in inventory");
        }
    }
```



```
}  
if (itemFound != true)  
{  
    Console.WriteLine("Item not found in the inventory");  
}  
break;
```

case 4:

```
    Console.Write("All the items list in inventory\n");  
    for (int i = 0; i < count; i++)  
    {  
        Console.WriteLine("Item {0} in inventory: {1}", i + 1, inventory[i]);  
    }  
    break;
```

default:

```
    Console.WriteLine("Invalid action selected");  
    break;
```

```
}
```

```
Console.Write("Do you want to end the program? (y/n): ");
```

```
exit = Convert.ToChar(Console.ReadLine());
```

```
}
```

```
while (exit == 'n' || exit == 'N');
```

```
}
```

OUTPUT :

```

Microsoft Visual Studio Debug Console
Specify the size of the inventory: 10
Enter the number of inventory items you want to setup? : 4

Enter the item name: tray
Enter the item name: roose
Enter the item name: keychain
Enter the item name: bag
Successful Inventory Setup

Action list:
1.Add Items
2.Remove Items
3.Search Items
4.Display Inventory
Select actions: 2

Enter the number of items you want to remove from the inventory: 1
Enter the item name: roose
Item removed successfully
Do you want to end the program? (y/n): n

Action list:
1.Add Items
2.Remove Items
3.Search Items
4.Display Inventory
Select actions: 1

Enter the number of items you want to add in the inventory: 2
Enter the item name: rose
Enter the item name: sunflower
Items added successfully
Do you want to end the program? (y/n): n

```

```

Microsoft Visual Studio Debug Console
Action List:
1.Add Items
2.Remove Items
3.Search Items
4.Display Inventory
Select actions: 3

Enter the item name: bag
Item found in inventory
Do you want to end the program? (y/n): n

Action List:
1.Add Items
2.Remove Items
3.Search Items
4.Display Inventory
Select actions: 3

Enter the item name: teddybear
Item not found in the inventory
Do you want to end the program? (y/n): n

Action List:
1.Add Items
2.Remove Items
3.Search Items
4.Display Inventory
Select actions: 4

All the items list in inventory
Item 1 in inventory: tray
Item 2 in inventory: keychain
Item 3 in inventory: bag
Item 4 in inventory: rose
Item 5 in inventory: sunflower
Do you want to end the program? (y/n): y

C:\Users\admin\OneDrive\Desktop\DSA\Array-DSA\1D-Arrays-2\bin\Debug\1D-Arrays-2.exe (process 21520) exited with code 0
Press any key to close this window . . .

```

TASK 3: Write a program which input 2 matrix of user defined rows and columns and perform following operation

- **Display/Print as a Matrix**
 - **Addition of Matrix**
 - **Subtraction of Matrix**
 - **matrix multiplication**
 - **Determinant**
 - **Inverse**
-

SOLUTION:

```
static void Main(string[] args)
{
    Console.Write("Enter the number of rows of 1st matrix: ");
    int mRow1 = Convert.ToInt32(Console.ReadLine());
    Console.Write("Enter the number of columnss of 1st matrix: ");
    int mCol1 = Convert.ToInt32(Console.ReadLine());
    Console.Write("Enter the number of rows of 2nd matrix: ");
    int mRow2 = Convert.ToInt32(Console.ReadLine());
    Console.Write("Enter the number of columns of 2nd matrix: ");
    int mCol2 = Convert.ToInt32(Console.ReadLine());

    int[,] matrix1 = new int[mRow1, mCol1];
    int[,] matrix2 = new int[mRow2, mCol2];
    int[,] addedMatrix = new int[mRow1, mCol1];

    char exit;
```

```
//VALUES OF MATRIX 1
```

```
for(int i = 0; i < mRow1; i++)
{
    for(int j = 0; j < mCol1; j++)
    {
        Console.Write("Enter the value at row {0} and column {1} of matrix 1 :", i + 1, j + 1);
        matrix1[i, j] = Convert.ToInt32(Console.ReadLine());
    }
}
Console.WriteLine("\n");
```

```
//VALUES OF MATRIX 2
```

```
for (int i = 0; i < mRow2; i++)
{
    for (int j = 0; j < mCol2; j++)
    {
        Console.Write("Enter the value at row {0} and column {1} of matrix 2 :", i + 1, j + 1);
        matrix2[i, j] = Convert.ToInt32(Console.ReadLine());
    }
}
```

```
do
```

```
{
    Console.WriteLine("\n");

    Console.Write("Action list: \n 1. Display Matrix \n 2. Addition of matrix \n 3. Subtraction of Matrix \n 4. Matrix multiplication \n 5. Determinant \n 6. Inverse \n Select action: ");

    int action = Convert.ToInt32(Console.ReadLine());
```

```
Console.WriteLine("\n");
switch (action)
{
    case 1:
        //DISPLAY OF MATRIX
        Console.Write("Select the matrix you want to display (1/2): ");
        int select = Convert.ToInt32(Console.ReadLine());
        if (select == 1)
        {
            Console.WriteLine("Matrix 1 :");
            for (int i = 0; i < mRow1; i++)
            {
                for (int j = 0; j < mCol1; j++)
                {
                    Console.Write(matrix1[i, j] + "\t");
                }
                Console.WriteLine();
            }
        }
        else if (select == 2)
        {
            Console.WriteLine("Matrix 2 :");
            for (int i = 0; i < mRow2; i++)
            {
                for (int j = 0; j < mCol2; j++)
                {
                    Console.Write(matrix2[i, j] + "\t");
                }
            }
        }
    }
}
```

```
        Console.WriteLine();  
    }  
}  
else  
{  
    Console.WriteLine("Matrix does not exist");  
}  
break;
```

case 2:

```
//ADDITION OF MATRIX  
if (mRow1 == mRow2 && mCol1 == mCol2)  
{  
    for (int i = 0; i < mRow1; i++)  
    {  
        for (int j = 0; j < mCol1; j++)  
        {  
            addedMatrix[i, j] = matrix1[i, j] + matrix2[i, j];  
            Console.Write(addedMatrix[i, j] + " ");  
        }  
        Console.WriteLine();  
    }  
}  
else  
{  
    Console.WriteLine("Only the matrix with same number of rows and columns can be  
added");  
}
```

```
break;
```

```
case 3:
```

```
//SUBTRACTION OF MATRIX
```

```
if (mRow1 == mRow2 && mCol1 == mCol2)
```

```
{
```

```
    for (int i = 0; i < mRow1; i++)
```

```
    {
```

```
        for (int j = 0; j < mCol1; j++)
```

```
        {
```

```
            addedMatrix[i, j] = matrix1[i, j] - matrix2[i, j];
```

```
            Console.Write(addedMatrix[i, j] + "\t");
```

```
        }
```

```
        Console.WriteLine();
```

```
    }
```

```
}
```

```
else
```

```
{
```

```
    Console.WriteLine("Only the matrix with same number of rows and columns can be  
added");
```

```
}
```

```
break;
```

```
case 4:
```

```
//MULTIPLICATION OF MATRIX
```

```
if (mCol1 == mRow2)
```

```
{
```

```
    int[,] multiply = new int[mRow1, mCol2];
```

```
for (int i = 0; i < mRow1; i++)
{
    for (int j = 0; j < mCol2; j++)
    {
        multiply[i, j] = 0;
        for (int k = 0; k < mCol1; k++)
        {
            multiply[i, j] += matrix1[i, k] * matrix2[k, j];
        }
        Console.Write(multiply[i, j] + "\t");
    }
    Console.WriteLine();
}
else
{
    Console.WriteLine("Matrix multiplication not possible");
}
break;
```

case 5:

```
//DETERMINANT OF MATRIX
```

```
Console.Write("Select the matrix you want to take determinant of (1/2): ");
```

```
select = Convert.ToInt32(Console.ReadLine());
```

```
int det;
```



```
if (select == 1)
{
    det = (matrix1[0, 0] * matrix1[1, 1]) - (matrix1[0, 1] * matrix1[1, 0]);
    Console.WriteLine("The determinant is {0}", det);
}
else if (select == 2)
{
    det = (matrix2[0, 0] * matrix2[1, 1]) - (matrix2[0, 1] * matrix2[1, 0]);
    Console.WriteLine("The determinant is {0}", det);
}
else
{
    Console.WriteLine("Matrix does not exist");
}
break;
```

case 6:

///INVERSE OF MATRIX

```
Console.Write("Select the matrix you want to take inverse of (1/2): ");
```

```
select = Convert.ToInt32(Console.ReadLine());
```

```
if (select == 1 && mRow1 == 2 && mCol1 == 2)
```

```
{
    det = (matrix1[0, 0] * matrix1[1, 1]) - (matrix1[0, 1] * matrix1[1, 0]);
    if (det == 0)
    {
        Console.WriteLine("Inverse not possible as determinant is 0");
    }
}
```

```
else
{
    double[,] inverse = new double[2, 2];
    inverse[0, 0] = matrix1[1, 1] / (double)det;
    inverse[0, 1] = -matrix1[0, 1] / (double)det;
    inverse[1, 0] = -matrix1[1, 0] / (double)det;
    inverse[1, 1] = matrix1[0, 0] / (double)det;

    Console.WriteLine("Inverse of matrix 1: ");
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            Console.Write(inverse[i, j] + "\t");
        }
        Console.WriteLine();
    }
}

else if (select == 2 && mRow2 == 2 && mCol2 == 2)
{
    det = (matrix2[0, 0] * matrix2[1, 1]) - (matrix2[0, 1] * matrix2[1, 0]);
    if (det == 0)
    {
        Console.WriteLine("Inverse not possible (determinant is 0).");
    }
}
```

```
else
{
    double[,] inverse = new double[2, 2];
    inverse[0, 0] = matrix2[1, 1] / (double)det;
    inverse[0, 1] = -matrix2[0, 1] / (double)det;
    inverse[1, 0] = -matrix2[1, 0] / (double)det;
    inverse[1, 1] = matrix2[0, 0] / (double)det;

    Console.WriteLine("Inverse of matrix 2:");
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            Console.Write(inverse[i, j] + "\t");
        }
        Console.WriteLine();
    }
}
else
{
    Console.WriteLine("Matrix does not exist.");
}
break;

default:
    Console.WriteLine("Invalid action");
    break;
```

```
}
```

```
Console.WriteLine("do you want to perform any other action? (y/n): ");
```

```
exit = Convert.ToChar(Console.ReadLine());
```

```
}
```

```
while(exit == 'n' || exit == 'N');
```

```
}
```

OUTPUT:

```

Microsoft Visual Studio Debug Console
Enter the number of rows of 1st matrix: 2
Enter the number of columnss of 1st matrix: 2
Enter the number of rows of 2nd matrix: 2
Enter the number of columns of 2nd matrix: 2
Enter the value at row 1 and column 1 of matrix 1 :2
Enter the value at row 1 and column 2 of matrix 1 :3
Enter the value at row 2 and column 1 of matrix 1 :5
Enter the value at row 2 and column 2 of matrix 1 :6

Enter the value at row 1 and column 1 of matrix 2 :4
Enter the value at row 1 and column 2 of matrix 2 :5
Enter the value at row 2 and column 1 of matrix 2 :7
Enter the value at row 2 and column 2 of matrix 2 :9

Action list:
1. Display Matrix
2. Addition of matrix
3. Subtraction of Matrix
4. Matrix multiplication
5. Determinant
6. Inverse
Select action: 1

Select the matrix you want to display (1/2): 1
Matrix 1 :
2      3
5      6
do you want to perform any other action? (y/n): n

Action list:
1. Display Matrix
2. Addition of matrix
3. Subtraction of Matrix
4. Matrix multiplication
5. Determinant
6. Inverse

```

```
Microsoft Visual Studio Debug Console
6      8
12     15
do you want to perform any other action? (y/n): n

Action list:
1. Display Matrix
2. Addition of matrix
3. Subtraction of Matrix
4. Matrix multiplication
5. Determinant
6. Inverse
Select action: 3

-2     -2
-2     -3
do you want to perform any other action? (y/n): n

Action list:
1. Display Matrix
2. Addition of matrix
3. Subtraction of Matrix
4. Matrix multiplication
5. Determinant
6. Inverse
Select action: 4

29     37
62     79
do you want to perform any other action? (y/n): n

Action list:
1. Display Matrix
2. Addition of matrix
3. Subtraction of Matrix
4. Matrix multiplication
5. Determinant
6. Inverse
Select action: 5
```

```
Select the matrix you want to take determinant of (1/2): 1
The determinant is -3
do you want to perform any other action? (y/n): n
```

```
Action list:
1. Display Matrix
2. Addition of matrix
3. Subtraction of Matrix
4. Matrix multiplication
5. Determinant
6. Inverse
Select action: 6
```

```
Select the matrix you want to take inverse of (1/2): 2
Inverse of matrix 2:
9      -5
-7     4
do you want to perform any other action? (y/n): y
```

```
C:\Users\admin\OneDrive\Desktop\DSA\Array-DSA\2D-Arrays-1\bin\Debug\2D-Arrays-1.exe (process 21
Press any key to close this window . . .|
```