

Data Structure lab assignment 1

Problem No: 1

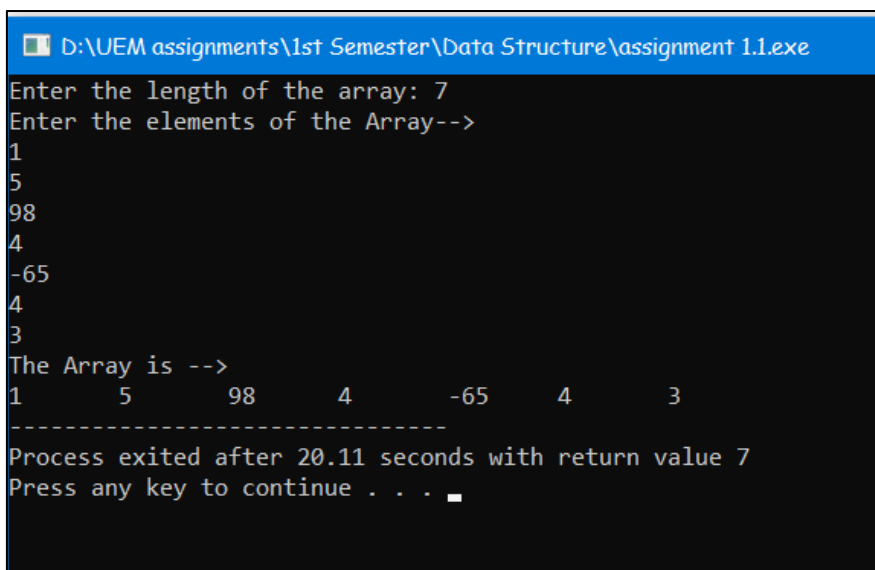
Problem Statement: Write a C program to print an array.

Source Code:

```
#include<stdio.h>

int main () {
    int n, i;
    printf("Enter the length of the array: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter the elements of the Array-->\n");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf("The Array is -->\n");
    for (i = 0; i < n; i++)
        printf("%d\t", arr[i]);
}
```

Output:



```
D:\UEM assignments\1st Semester\Data Structure\assignment 1.1.exe
Enter the length of the array: 7
Enter the elements of the Array-->
1
5
98
4
-65
4
3
The Array is -->
1      5      98      4      -65      4      3
-----
Process exited after 20.11 seconds with return value 7
Press any key to continue . . .
```

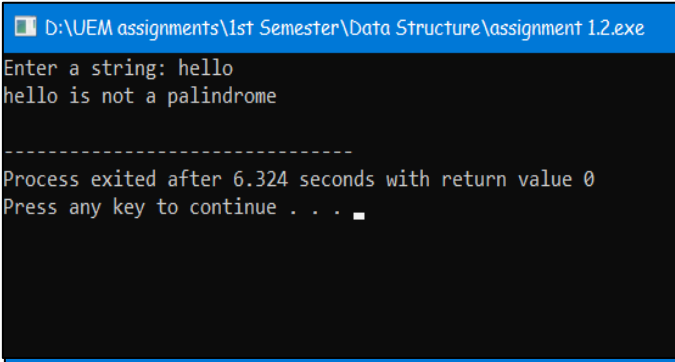
Problem No: 2

Problem Statement: Write a C program to check whether a given string is Palindrome or not.

Source Code:

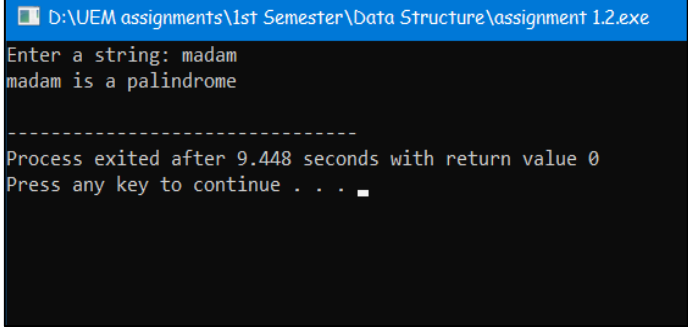
```
#include<stdio.h>
#include<string.h>
int main() {
    char str[100];
    int l = 0, h;
    printf("Enter a string: ");
    gets(str);
    h = strlen(str) - 1;
    while (h > l) {
        if (str[l++] != str[h--]) {
            printf("%s is not a palindrome\n", str);
            return 0;
        }
    }
    printf("%s is a palindrome\n", str);
    return 0;
}
```

Output:



```
D:\UEM assignments\1st Semester\Data Structure\assignment 1.2.exe
Enter a string: hello
hello is not a palindrome

-----
Process exited after 6.324 seconds with return value 0
Press any key to continue . . .
```



```
D:\UEM assignments\1st Semester\Data Structure\assignment 1.2.exe
Enter a string: madam
madam is a palindrome

-----
Process exited after 9.448 seconds with return value 0
Press any key to continue . . .
```

Problem No: 3

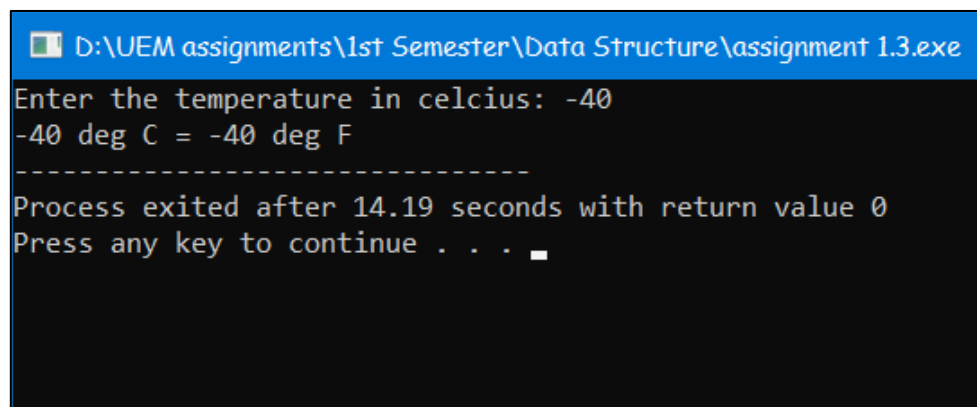
Problem Statement: Write a C program to convert temperature from degree Centigrade to Fahrenheit.

Source Code:

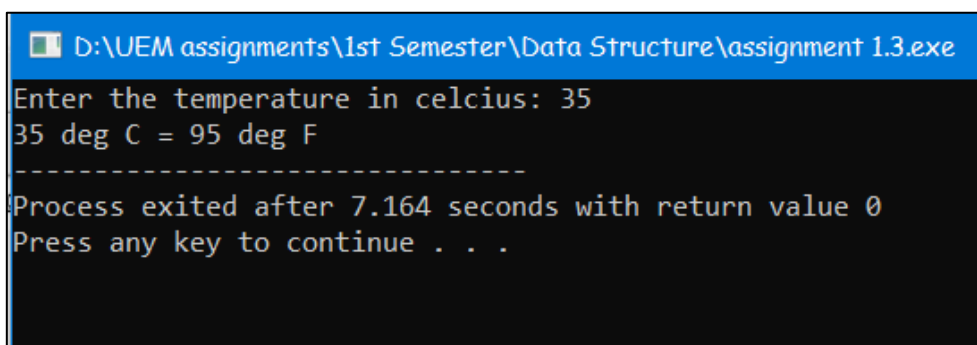
```
#include<stdio.h>

int main () {
    int tc, tf;
    printf("Enter the temperature in celcius: ");
    scanf("%d", &tc);
    tf = (tc * 9 / 5) + 32;
    printf("%d deg C = %d deg F", tc, tf);
    return 0;
}
```

Output:



```
D:\UEM assignments\1st Semester\Data Structure\assignment 1.3.exe
Enter the temperature in celcius: -40
-40 deg C = -40 deg F
-----
Process exited after 14.19 seconds with return value 0
Press any key to continue . . .
```



```
D:\UEM assignments\1st Semester\Data Structure\assignment 1.3.exe
Enter the temperature in celcius: 35
35 deg C = 95 deg F
-----
Process exited after 7.164 seconds with return value 0
Press any key to continue . . .
```

Problem No: 4

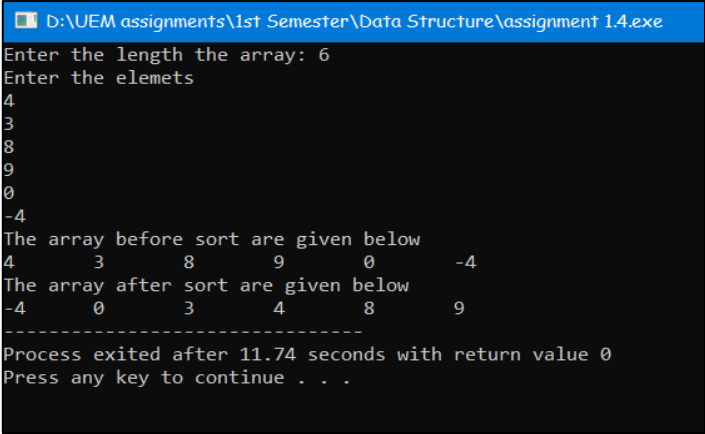
Problem Statement: Write a C program to sort an array.

Source Code:

```
#include <stdio.h>

int main() {
    int i, j, temp, len;
    printf("Enter the length the array: ");
    scanf("%d", &len);
    int arr[len];
    printf("Enter the elemets \n");
    for (i = 0; i < len; i++)
        scanf("%d", &arr[i]);
    printf("The array before sort are given below \n");
    for (i = 0; i < len; i++)
        printf("%d\t", arr[i]);
    for (i = 0; i < len; i++)
        for (j = i + 1; j < len; j++)
            if (arr[i] > arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
    printf("\nThe array after sort are given below \n");
    for (i = 0; i < len; i++)
        printf("%d\t", arr[i]);
    return 0;
}
```

Output:



```
D:\UEM assignments\1st Semester\Data Structure\assignment 1.4.exe
Enter the length the array: 6
Enter the elemets
4
3
8
9
0
-4
The array before sort are given below
4      3      8      9      0      -4
The array after sort are given below
-4      0      3      4      8      9
-----
Process exited after 11.74 seconds with return value 0
Press any key to continue . . .
```

Problem No: 5

Problem Statement: Write a C program to print the largest and second largest element of the array.

Source Code:

```
#include <stdio.h>

int main() {
    int n, max, max2, i, has_max2 = 0;
    printf("Enter the length of the array \n");
    scanf("%d", &n);
    int arr[n];
    printf("Enter the elements \n");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf ("The array is->\n");
    for (i = 0; i < n; i++)
        printf("%d\t", arr[i]);
    max = arr[0];
    for (i = 0; i < n; i++) {
        if (max < arr[i]) {
            max2 = max;
            max = arr[i];
        }
    }
    for (i = 0; i < n; i++) {
        if (arr[i] < max) {
            if (!has_max2) {
                has_max2 = 1;
                max2 = arr[i];
            }
            else if (arr[i] > max2)
                max2 = arr[i];
        }
    }
    if (has_max2 == 1)
```

```

        printf("\nLargest number = %d\n2nd Largest number = %d", max,
max2);
    else
        printf("\nAll values are identical to %d", max);
    return 0;
}

```

Output:

```

D:\UEM assignments\1st Semester\Data Structure\assignment 1.5.exe
Enter the length of the array
6
Enter the elements
9
6
4
8
-5
0
The array is->
9      6      4      8      -5      0
Largest number = 9
2nd Largest number = 8
-----
Process exited after 14.56 seconds with return value 0
Press any key to continue . . .

```

```

D:\UEM assignments\1st Semester\Data Structure\assignment 1.5.exe
Enter the length of the array
5
Enter the elements
8
4
7
4
2
The array is->
8      4      7      4      2
Largest number = 8
2nd Largest number = 7
-----
Process exited after 30.78 seconds with return value 0
Press any key to continue . . .

```

Problem No: 6

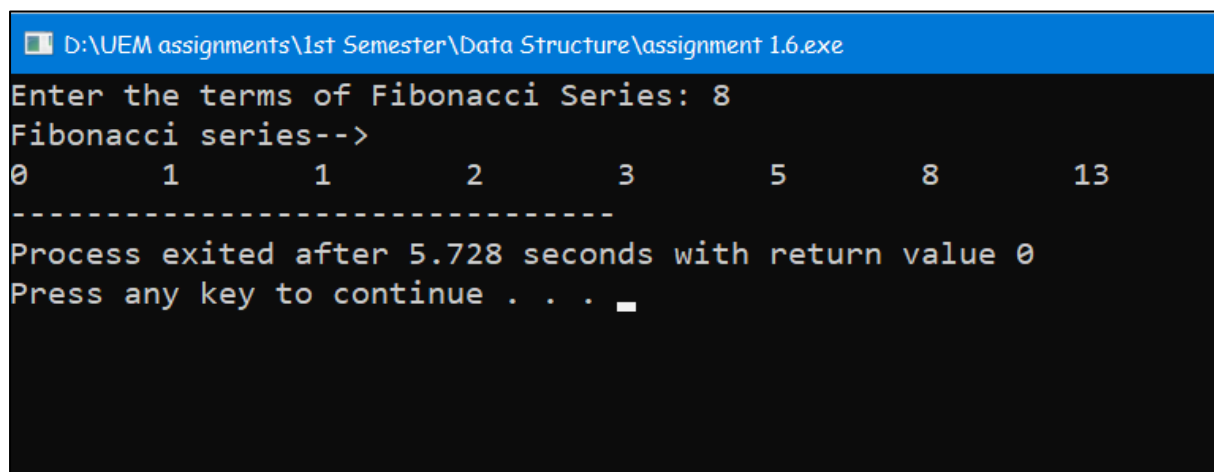
Problem Statement: Write a C program to display Fibonacci series.

Source Code:

```
#include <stdio.h>

int main() {
    int num, i, t1 = 0, t2 = 1, next = t1 + t2;
    printf("Enter the terms of Fibonacci Series: ");
    scanf("%d", &num);
    printf("Fibonacci series-->\n");
    printf("%d\t%d\t", t1, t2);
    for (i = 2; i < num; i++) {
        printf("%d\t", next);
        t1 = t2;
        t2 = next;
        next = t1 + t2;
    }
    return 0;
}
```

Output:



The screenshot shows a Windows command prompt window with the title bar "D:\UEM assignments\1st Semester\Data Structure\assignment 1.6.exe". The program prompts the user to "Enter the terms of Fibonacci Series: 8". It then displays the Fibonacci series: "Fibonacci series-->" followed by the numbers 0, 1, 1, 2, 3, 5, 8, and 13, each separated by a tab character. A dashed line is printed below the series. The program then displays "Process exited after 5.728 seconds with return value 0" and "Press any key to continue . . .".

Problem No: 7

Problem Statement: Write a program that reads two 2D metrics from the console, verifies if metrics multiplication is possible or not. Then multiplies the metrics and prints the 3rd metrics.

Source Code:

```
#include<stdio.h>

int main() {
    int row1, row2, column1, column2, i, j, k;
    printf("Enter the row and column of the 1st Matrix-->\n");
    printf("Row: ");
    scanf("%d", &row1);
    printf("Column: ");
    scanf("%d", &column1);
    printf("Enter the row and column of the 2nd Matrix-->\n");
    printf("Row: ");
    scanf("%d", &row2);
    printf("Column: ");
    scanf("%d", &column2);
    if (column1 != row2) {
        printf("1st matrix columns is not equal to 2nd matrix\nrow.\nMultiplication Can't possible.");
        return 0;
    }
    int matrix1 [row1][column1], matrix2 [row2][column2], result [row1][column2];
    printf("Enter the elements of 1st Matrix-->\n");
    for (i = 0; i < row1; ++i)
        for (j = 0; j < column1; ++j) {
            printf("Enter element at [%d] [%d]: ", i + 1, j + 1);
            scanf("%d", &matrix1[i][j]);
        }
    printf("Enter the elements of 2nd Matrix-->\n");
    for (i = 0; i < row2; ++i)
        for (j = 0; j < column2; ++j) {
            printf("Enter element at [%d] [%d]: ", i + 1, j + 1);
            scanf("%d", &matrix2[i][j]);
        }
}
```



```

    }

    for (i = 0; i < row1; ++i)
        for (j = 0; j < column2; ++j)
            result[i][j] = 0;
    for (i = 0; i < row1; ++i)
        for (j = 0; j < column2; ++j)
            for (k = 0; k < column1; ++k)
                result[i][j] += matrix1[i][k] * matrix2[k][j];
    printf("Multiplication of two matrices is-->\n");

    for (i = 0; i < row1; i++) {
        for (j = 0; j < column2; j++)
            printf("%d  ", result[i][j]);

        printf("\n");
    }

    return 0;
}

```

Output:

```

D:\UEM assignments\1st Semester\Data Structure\assignment 1.7.exe
Enter the row and column of the 1st Matrix-->
Row: 2
Column: 3
Enter the row and column of the 2nd Matrix-->
Row: 3
Column: 3
Enter the elements of 1st Matrix-->
Enter element at [1] [1]: 1
Enter element at [1] [2]: 2
Enter element at [1] [3]: 3
Enter element at [2] [1]: 4
Enter element at [2] [2]: 5
Enter element at [2] [3]: 6
Enter the elements of 2nd Matrix-->
Enter element at [1] [1]: 9
Enter element at [1] [2]: 8
Enter element at [1] [3]: 7
Enter element at [2] [1]: 6
Enter element at [2] [2]: 6
Enter element at [2] [3]: 5
Enter element at [3] [1]: 4
Enter element at [3] [2]: 3
Enter element at [3] [3]: 2
Multiplication of two matrices is-->
33 29 23
90 80 65

-----
Process exited after 41.44 seconds with return value 0
Press any key to continue . . .

```

```

D:\UEM assignments\1st Semester\Data Structure\assignment 1.7.exe
Enter the row and column of the 1st Matrix-->
Row: 3
Column: 3
Enter the row and column of the 2nd Matrix-->
Row: 2
Column: 2
1st matrix columns is not equal to 2nd matrix row.
Multiplication Can't possible.

-----
Process exited after 9.324 seconds with return value 0
Press any key to continue . . .

```

Problem No: 8

Problem Statement: Write a program that reads a 2D metrics and checks if the metrics is a symmetric metrics or not.

Source Code:

```
#include<stdio.h>

int main() {
    int row, column, i, j, flag = 0;
    printf("Enter the row and column of the Matrix-->\n");
    printf("Row: ");
    scanf("%d", &row);
    printf("Column: ");
    scanf("%d", &column);
    int matrix [row][column];
    printf("Enter the elements of the Matrix-->\n");
    for (i = 0; i < row; i++)
        for (j = 0; j < column; j++) {
            printf("Enter element at [%d] [%d]: ", i + 1, j + 1);
            scanf("%d", &matrix[i][j]);
        }
    for (i = 0; i < row; i++)
        for (j = 0; j < column; j++)
            if (matrix [j][i] != matrix [i][j]) {
                flag = 1;
                break;
            }

    if (flag == 0)
        printf("The matrix is a symmetric matrix.");
    else
        printf("The matrix is not a symmetric matrix.");
    return 0;
}
```

Output:

```
D:\UEM assignments\1st Semester\Data Structure\assignment 1.8.exe
Enter the row and column of the Matrix-->
Row: 3
Column: 3
Enter the elements of the Matrix-->
Enter element at [1] [1]: 1
Enter element at [1] [2]: 0
Enter element at [1] [3]: -1
Enter element at [2] [1]: 0
Enter element at [2] [2]: 5
Enter element at [2] [3]: 8
Enter element at [3] [1]: -1
Enter element at [3] [2]: 8
Enter element at [3] [3]: 6
The matrix is a symmetric matrix.
-----
Process exited after 61.6 seconds with return value 0
Press any key to continue . . .
```

```
D:\UEM assignments\1st Semester\Data Structure\assignment 1.8.exe
Enter the row and column of the Matrix-->
Row: 3
Column: 3
Enter the elements of the Matrix-->
Enter element at [1] [1]: 1
Enter element at [1] [2]: 2
Enter element at [1] [3]: 3
Enter element at [2] [1]: 4
Enter element at [2] [2]: 5
Enter element at [2] [3]: 6
Enter element at [3] [1]: 7
Enter element at [3] [2]: 8
Enter element at [3] [3]: 9
The matrix is not a symmetric matrix.
-----
Process exited after 14.08 seconds with return value 0
Press any key to continue . . .
```

Problem No: 9

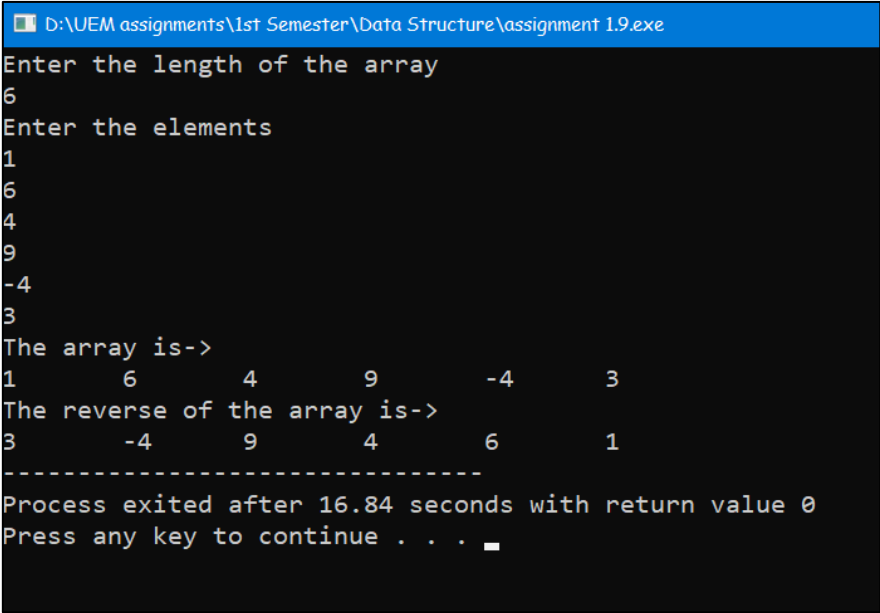
Problem Statement: Write a C program to print reverse array.

Source Code:

```
#include<stdio.h>

int main() {
    int len, i;
    printf("Enter the length of the array \n");
    scanf("%d", &len);
    int arr[len];
    printf("Enter the elements \n");
    for (i = 0; i < len; i++)
        scanf("%d", &arr[i]);
    printf ("The array is->\n");
    for (i = 0; i < len; i++)
        printf("%d\t", arr[i]);
    printf ("\nThe reverse of the array is->\n");
    for (i = len - 1; i >= 0; i--)
        printf("%d\t", arr[i]);
    return 0;
}
```

Output:



```
D:\UEM assignments\1st Semester\Data Structure\assignment 1.9.exe
Enter the length of the array
6
Enter the elements
1
6
4
9
-4
3
The array is->
1      6      4      9      -4      3
The reverse of the array is->
3      -4      9      4      6      1
-----
Process exited after 16.84 seconds with return value 0
Press any key to continue . . .
```

Problem No: 10

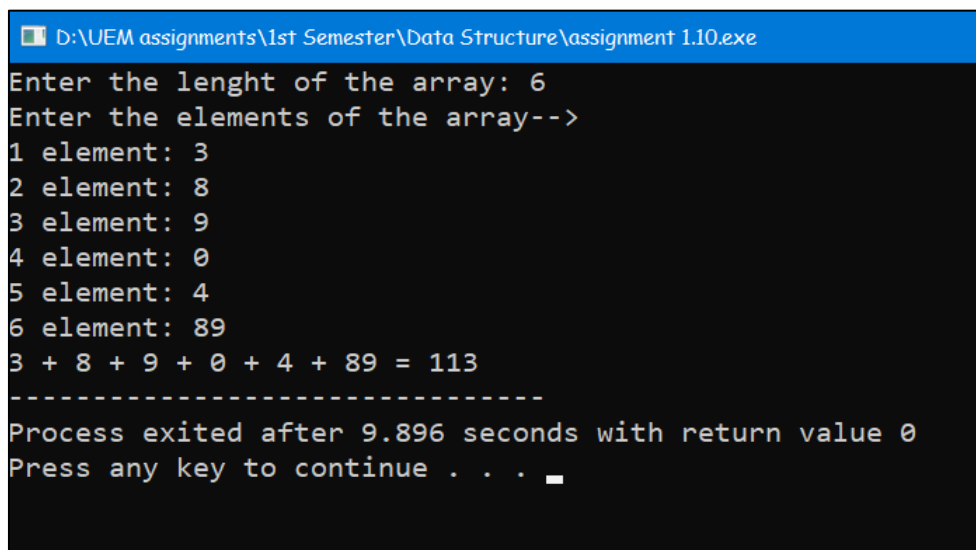
Problem Statement: Write a C program to check the sum of all elements of an array.

Source Code:

```
#include<stdio.h>

int main() {
    int len, i, sum = 0;
    printf("Enter the lenght of the array: ");
    scanf("%d", &len);
    int arr[len];
    printf("Enter the elements of the array-->\n");
    for (i = 0; i < len; i++) {
        printf("%d element: ", i + 1);
        scanf("%d", &arr[i]);
    }
    for (i = 0; i < len; i++) {
        sum += arr[i];
        printf("%d + ", arr[i]);
    }
    printf("\b\b= %d", sum);
    return 0;
}
```

Output:



```
D:\UEM assignments\1st Semester\Data Structure\assignment 1.10.exe
Enter the lenght of the array: 6
Enter the elements of the array-->
1 element: 3
2 element: 8
3 element: 9
4 element: 0
5 element: 4
6 element: 89
3 + 8 + 9 + 0 + 4 + 89 = 113
-----
Process exited after 9.896 seconds with return value 0
Press any key to continue . . .
```

Problem No: 11

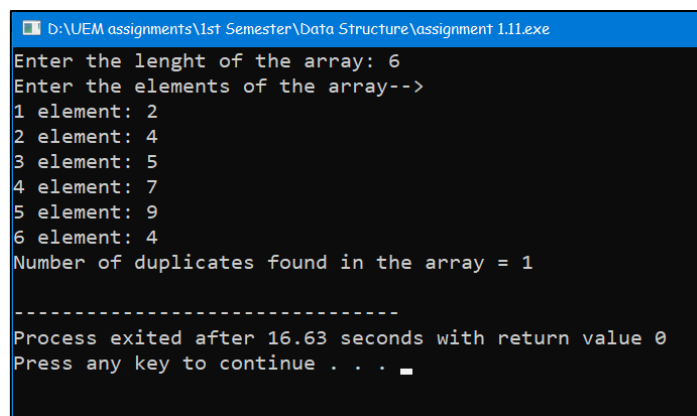
Problem Statement: Write a C program to check duplicate number in an array.

Source Code:

```
#include <stdio.h>

int main() {
    int len, i, j, count = 0;
    printf("Enter the lenght of the array: ");
    scanf("%d", &len);
    int arr[len];
    printf("Enter the elements of the array-->\n");
    for (i = 0; i < len; i++) {
        printf("%d element: ", i + 1);
        scanf("%d", &arr[i]);
    }
    for (i = 0; i < len; i++)
        for (j = i + 1; j < len; j++)
            if (arr[i] == arr[j]) {
                count++;
                break;
            }
    if (count == 0)
        printf("No duplicates found in the array.\n");
    else
        printf("Number of duplicates found in the array = %d\n",
count);
    return 0;
}
```

Output:

A screenshot of a Windows command prompt window with a blue title bar. The title bar text is "D:\UEM assignments\1st Semester\Data Structure\assignment 11.exe". The command prompt shows the following text: "Enter the lenght of the array: 6", "Enter the elements of the array-->", "1 element: 2", "2 element: 4", "3 element: 5", "4 element: 7", "5 element: 9", "6 element: 4", "Number of duplicates found in the array = 1", a separator line of dashes, "Process exited after 16.63 seconds with return value 0", and "Press any key to continue . . .".

```
D:\UEM assignments\1st Semester\Data Structure\assignment 11.exe
Enter the lenght of the array: 6
Enter the elements of the array-->
1 element: 2
2 element: 4
3 element: 5
4 element: 7
5 element: 9
6 element: 4
Number of duplicates found in the array = 1
-----
Process exited after 16.63 seconds with return value 0
Press any key to continue . . .
```