ARRAYS

SARTHAK SANAY

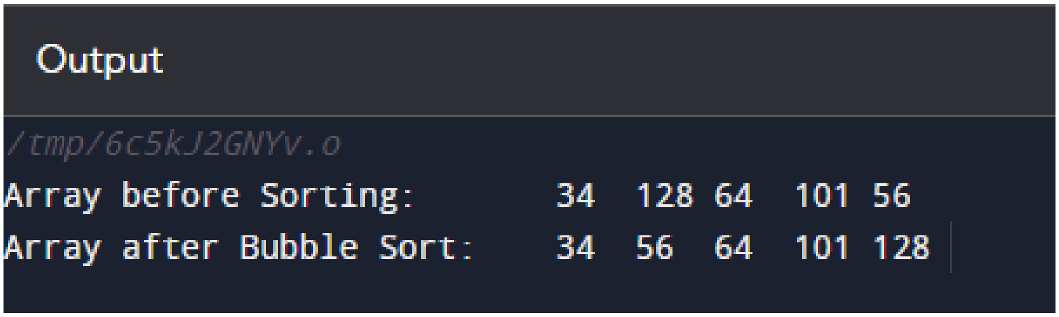
(1) AIM:-

To write programs in C to manipulate arrays (sorting, searching, insertion, deletion)

CODE 1:- (Sorting -> Bubble Sort)

```
// Program in C to implement Sorting (Bubble Sort) in an array
#include <stdio.h>
int main()
{
    int arr[5]= {34,128,64,101,56};
    printf("Array before Sorting: \t\t");
    for(int i=0; i<5; i++)
    {
        printf("%d\t", arr[i]);
    }
    // Implementing Ascending Bubble Sort
    for(int i=0; i<5; i++)
    {
        for(int j=0; j<5-i-1; j++)
        {
            if(arr[j] > arr[j+1])
            {
                int temp= arr[j];
                arr[j]= arr[j+1];
                arr[j+1]= temp;
            }
        }
    }
    printf("\nArray after Bubble Sort: \t");
    for(int i=0; i<5; i++)
    {
        printf("%d\t", arr[i]);
    }
    return 0;
```

OUTPUT SCREEN 1:-

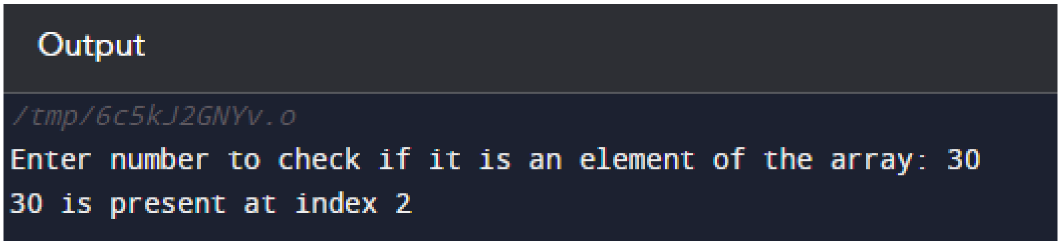


```
/tmp/6c5kJ2GNYv.o
Array before Sorting:       34   128 64   101  56
Array after Bubble Sort:    34   56   64   101  128
```

CODE 2:- (Searching -> Linear Search)

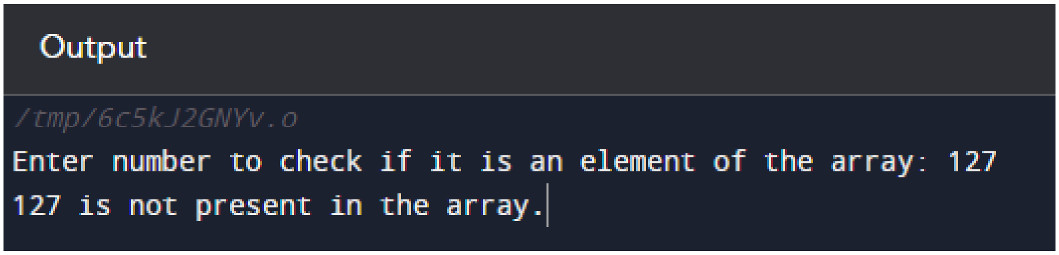
```
// Program in C to implement Linear searching in an array
#include <stdio.h>
int main()
{
    int c=0, n;
    int arr[5]= {10,20,30,40,50};
    printf("Enter number to check if it is an element of the array: ");
    scanf("%d", &n);
    for(int i=0; i<5; i++)
    {
        if(arr[i] == n)
        {
            printf("%d is present at index %d",n,i);
            c=1;
        }
    }
    if(c==0)
        printf("%d is not present in the array.", n);
    return 0;
}
```

OUTPUT SCREEN 2:-



Output

```
/tmp/6c5kJ2GNYv.o
Enter number to check if it is an element of the array: 30
30 is present at index 2
```



Output

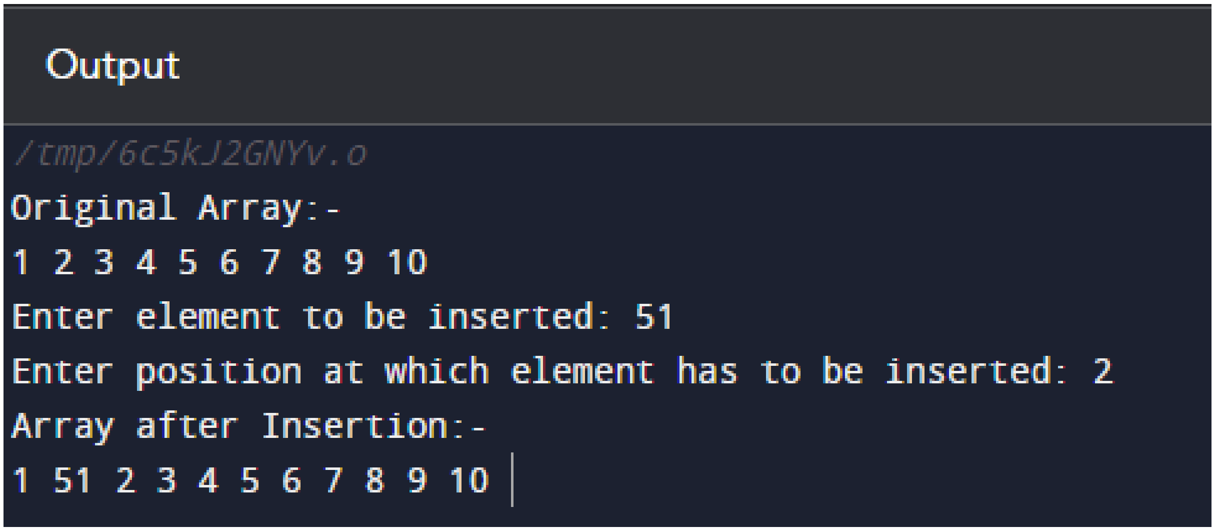
```
/tmp/6c5kJ2GNYv.o
Enter number to check if it is an element of the array: 127
127 is not present in the array.
```

P.T.O.

CODE 3:- (Insertion of element)

```
// Program in C to implement Insertion of an element at a specific position in an array
#include <stdio.h>
int main()
{
    int arr[100]= {1,2,3,4,5,6,7,8,9,10};
    int x, pos, n=10;
    printf("Original Array:-\n");
    for (int i=0; i<n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\nEnter element to be inserted: ");
    scanf("%d", &x);
    printf("Enter position at which element has to be inserted: ");
    scanf("%d", &pos);
    n++; // increase the size by 1
    // shifting elements forward
    for (int i=n-1; i>=pos; i--)
    {
        arr[i] = arr[i-1];
    }
    arr[pos-1] = x; // inserting elements at pos
    printf("Array after Insertion:-\n");
    for (int i=0; i<n; i++)
    {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

OUTPUT SCREEN 3:-



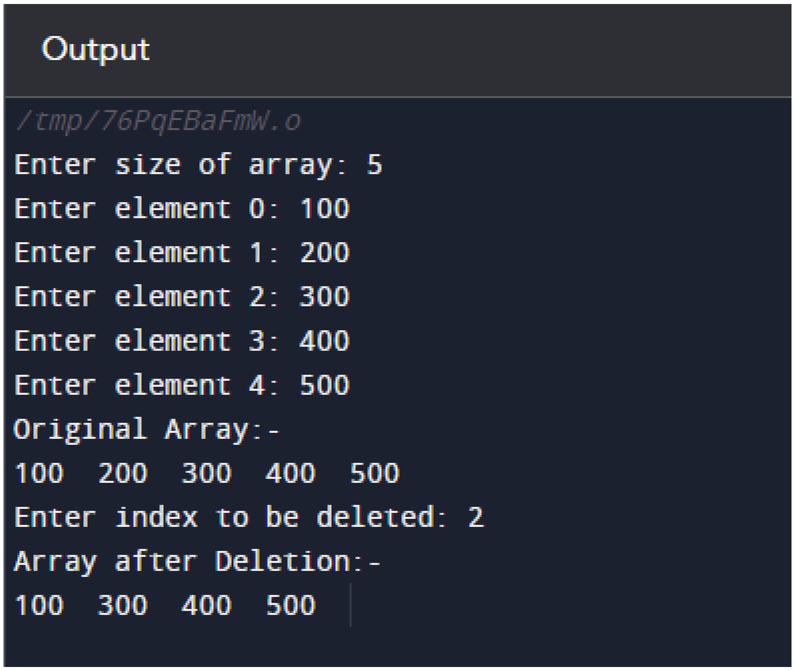
```
/tmp/6c5kJ2GNYv.o
Original Array:-
1 2 3 4 5 6 7 8 9 10
Enter element to be inserted: 51
Enter position at which element has to be inserted: 2
Array after Insertion:-
1 51 2 3 4 5 6 7 8 9 10
```

P.T.O.

CODE 4:- (Deletion of element)

```c
// Program in C to implement Deletion of an element in an Array from the desired position
#include <stdio.h>
int main()
{
    int n, p;
    printf("Enter size of array: ");
    scanf("%d", &n);
    int arr[n];
    for(int i=0; i<n; i++)
    {
        printf("Enter element %d: ", i);
        scanf("%d", &arr[i]);
    }
    printf("Original Array:-\n");
    for(int i=0; i<n; i++)
    {
        printf("%d  ", arr[i]);
    }
    printf("\nEnter index to be deleted: ");
    scanf("%d", &p);
    if (p >= n+1)
        printf("Deletion not possible. Enter valid index position.\n");
    else
    {
        for(int i=(p-1); i<(n-1); i++)
        {
            arr[i]= arr[i+1];
        }
        printf("Array after Deletion:- \n");
        // for( c = 0 ; c < n - 1 ; c++ )
        for(int i=0; i<n-1; i++)
        {
            printf("%d  ", arr[i]);
        }
    }
    return 0;
}
```

OUTPUT SCREEN 4:-



```
/tmp/76PqEBaFmW.o
Enter size of array: 5
Enter element 0: 100
Enter element 1: 200
Enter element 2: 300
Enter element 3: 400
Enter element 4: 500
Original Array:-
100  200  300  400  500
Enter index to be deleted: 2
Array after Deletion:-
100  300  400  500
```

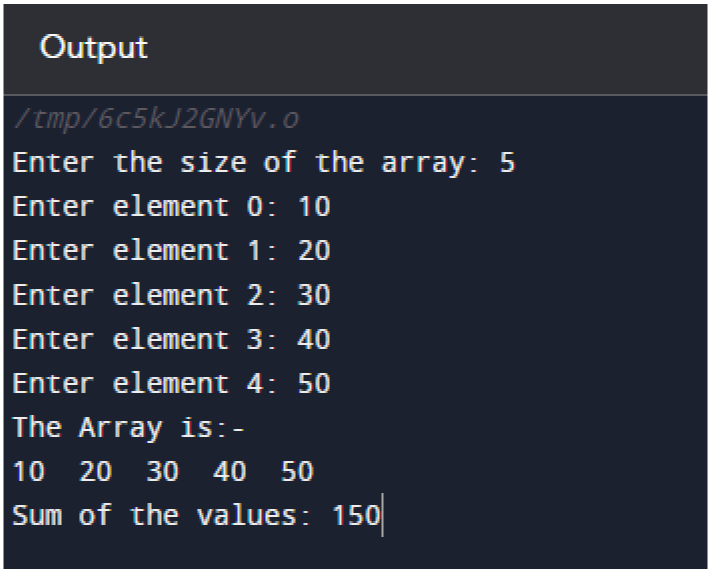
(2) AIM:-

To write programs in C to find the sum, average, maximum, and minimum values in an array.

CODE 1:- (Sum)

```
// Program in C to find the sum of the values in an array
#include <stdio.h>
int main()
{
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int arr[n];
    for(int i=0; i<n; i++)
    {
        printf("Enter element %d: ",i);
        scanf("%d", &arr[i]);
    }
    printf("The Array is:-\n");
    for(int i=0; i<n; i++)
    {
        printf("%d  ", arr[i]);
    }
    int sum= 0;
    for(int i=0; i<n; i++)
    {
        sum+= arr[i];
    }
    printf("\nSum of the values: %d", sum);
    return 0;
}
```

OUTPUT SCREEN 1:-

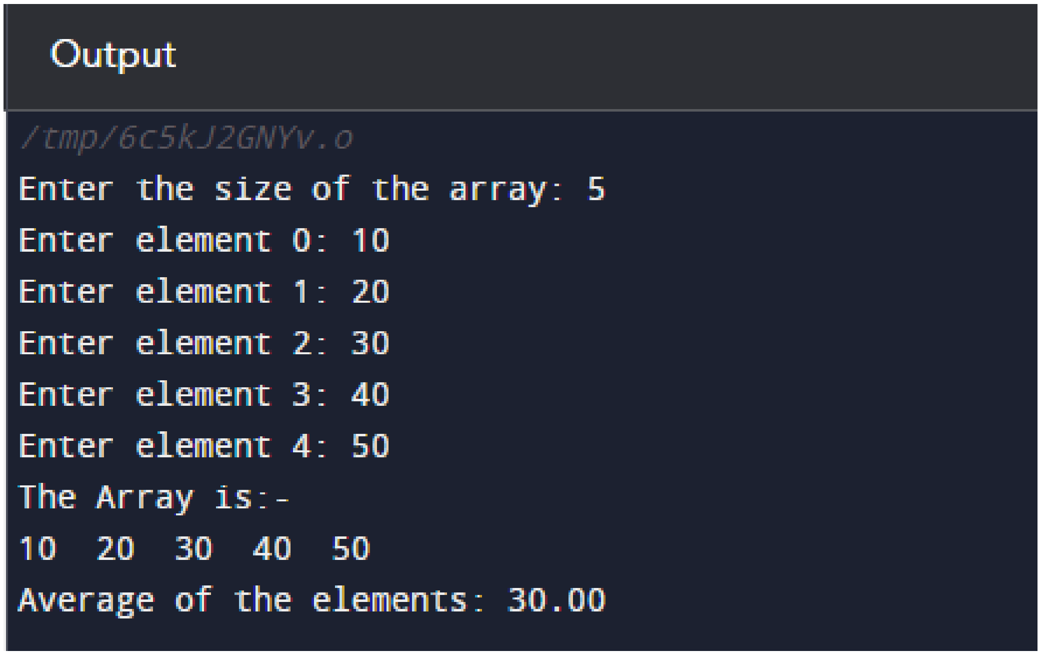


```
/tmp/6c5kJ2GNYv.o
Enter the size of the array: 5
Enter element 0: 10
Enter element 1: 20
Enter element 2: 30
Enter element 3: 40
Enter element 4: 50
The Array is:-
10  20  30  40  50
Sum of the values: 150
```

CODE 2:- (Average)

```
// Program in C to find the average of the values in an array
#include <stdio.h>
int main()
{
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int arr[n];
    for(int i=0; i<n; i++)
    {
        printf("Enter element %d: ",i);
        scanf("%d", &arr[i]);
    }
    printf("The Array is:-\n");
    for(int i=0; i<n; i++)
    {
        printf("%d  ", arr[i]);
    }
    int sum= 0;
    for(int i=0; i<n; i++)
    {
        sum+= arr[i];
    }
    double avg= (double)sum/n;
    printf("\nAverage of the elements: %.2f", avg);
    return 0;
}
```

OUTPUT SCREEN 2:-

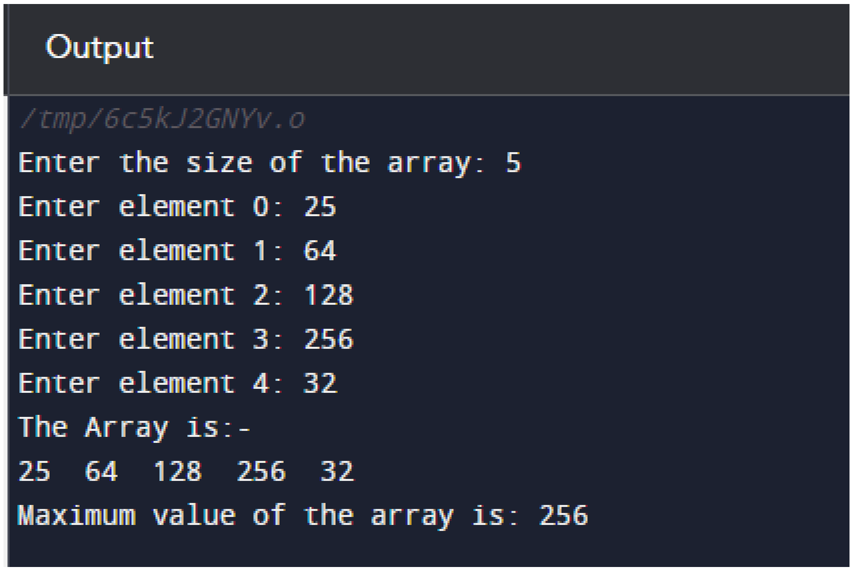


```
/tmp/6c5kJ2GNYv.o
Enter the size of the array: 5
Enter element 0: 10
Enter element 1: 20
Enter element 2: 30
Enter element 3: 40
Enter element 4: 50
The Array is:-
10  20  30  40  50
Average of the elements: 30.00
```

CODE 3:- (Maximum)

```c
// Program in C to find the maximum element in an array
#include <stdio.h>
int main()
{
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int arr[n];
    for(int i=0; i<n; i++)
    {
        printf("Enter element %d: ",i);
        scanf("%d", &arr[i]);
    }
    printf("The Array is:-\n");
    for(int i=0; i<n; i++)
    {
        printf("%d  ", arr[i]);
    }
    int max= arr[0];
    for(int i=0; i<n; i++)
    {
        if(arr[i]>max)
            max= arr[i];
    }
    printf("\nMaximum value of the array is: %d", max);
    return 0;
}
```

OUTPUT SCREEN 3:-

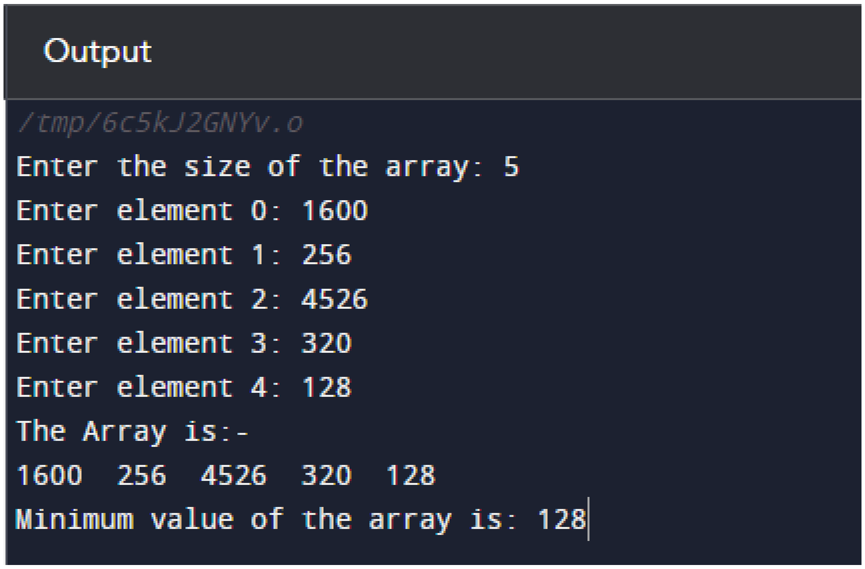


```
/tmp/6c5kJ2GNYv.o
Enter the size of the array: 5
Enter element 0: 25
Enter element 1: 64
Enter element 2: 128
Enter element 3: 256
Enter element 4: 32
The Array is:-
25  64  128  256  32
Maximum value of the array is: 256
```

CODE 4:- (Minimum)

```c
// Program in C to find the minimum element in an array
#include <stdio.h>
int main()
{
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int arr[n];
    for(int i=0; i<n; i++)
    {
        printf("Enter element %d: ",i);
        scanf("%d", &arr[i]);
    }
    printf("The Array is:-\n");
    for(int i=0; i<n; i++)
    {
        printf("%d  ", arr[i]);
    }
    int min= arr[0];
    for(int i=0; i<n; i++)
    {
        if(arr[i]<min)
            min= arr[i];
    }
    printf("\nMinimum value of the array is: %d", min);
    return 0;
}
```

OUTPUT SCREEN 4:-



```
/tmp/6c5kJ2GNYv.o
Enter the size of the array: 5
Enter element 0: 1600
Enter element 1: 256
Enter element 2: 4526
Enter element 3: 320
Enter element 4: 128
The Array is:-
1600  256  4526  320  128
Minimum value of the array is: 128
```

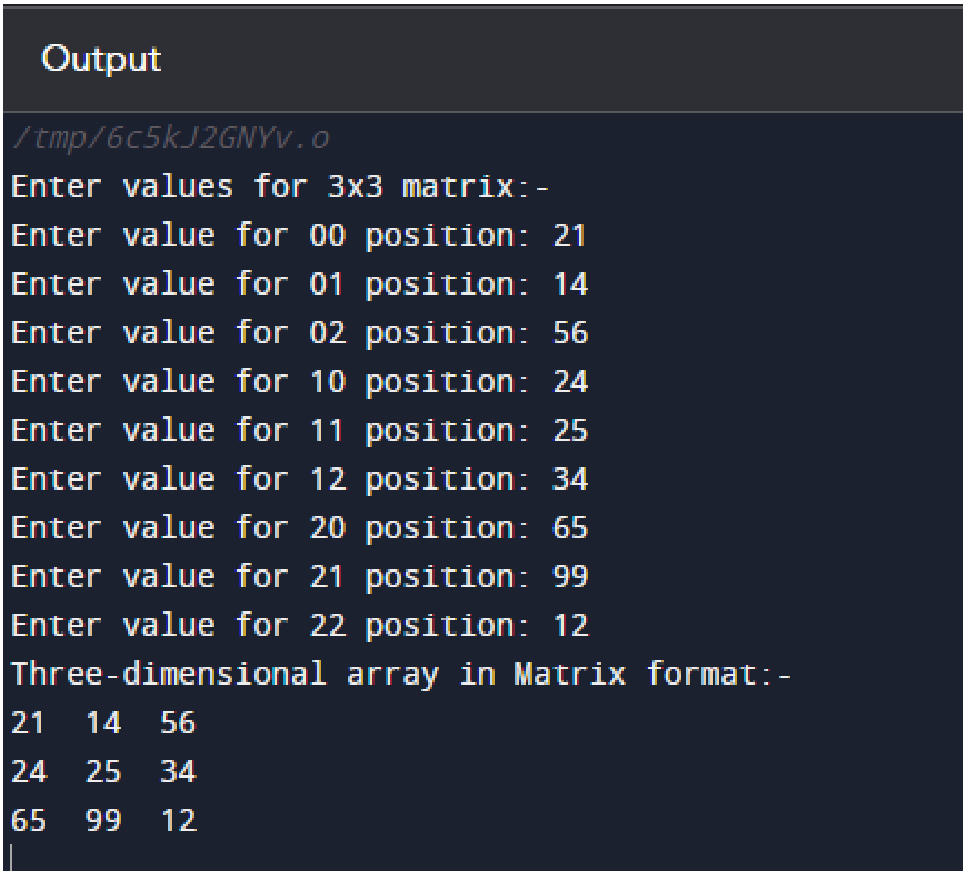
(3) AIM:-

To practice matrix operations in C using multi-dimensional arrays.

CODE 1:- (Display 3D array as matrix)

```
// Program in C to display a three-dimensional array as a matrix
#include <stdio.h>
int main()
{
    int m[3][3];
    printf("Enter values for 3x3 matrix:-\n");
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("Enter value for %d%d position: ", i,j);
            scanf("%d", &m[i][j]);
        }
    }
    printf("Three-dimensional array in Matrix format:-\n");
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("%d  ", m[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT SCREEN 1:-



```
/tmp/6c5kJ2GNYv.o
Enter values for 3x3 matrix:-
Enter value for 00 position: 21
Enter value for 01 position: 14
Enter value for 02 position: 56
Enter value for 10 position: 24
Enter value for 11 position: 25
Enter value for 12 position: 34
Enter value for 20 position: 65
Enter value for 21 position: 99
Enter value for 22 position: 12
Three-dimensional array in Matrix format:-
21  14  56
24  25  34
65  99  12
```

P.T.O.

CODE 2:- (Matrix addition)

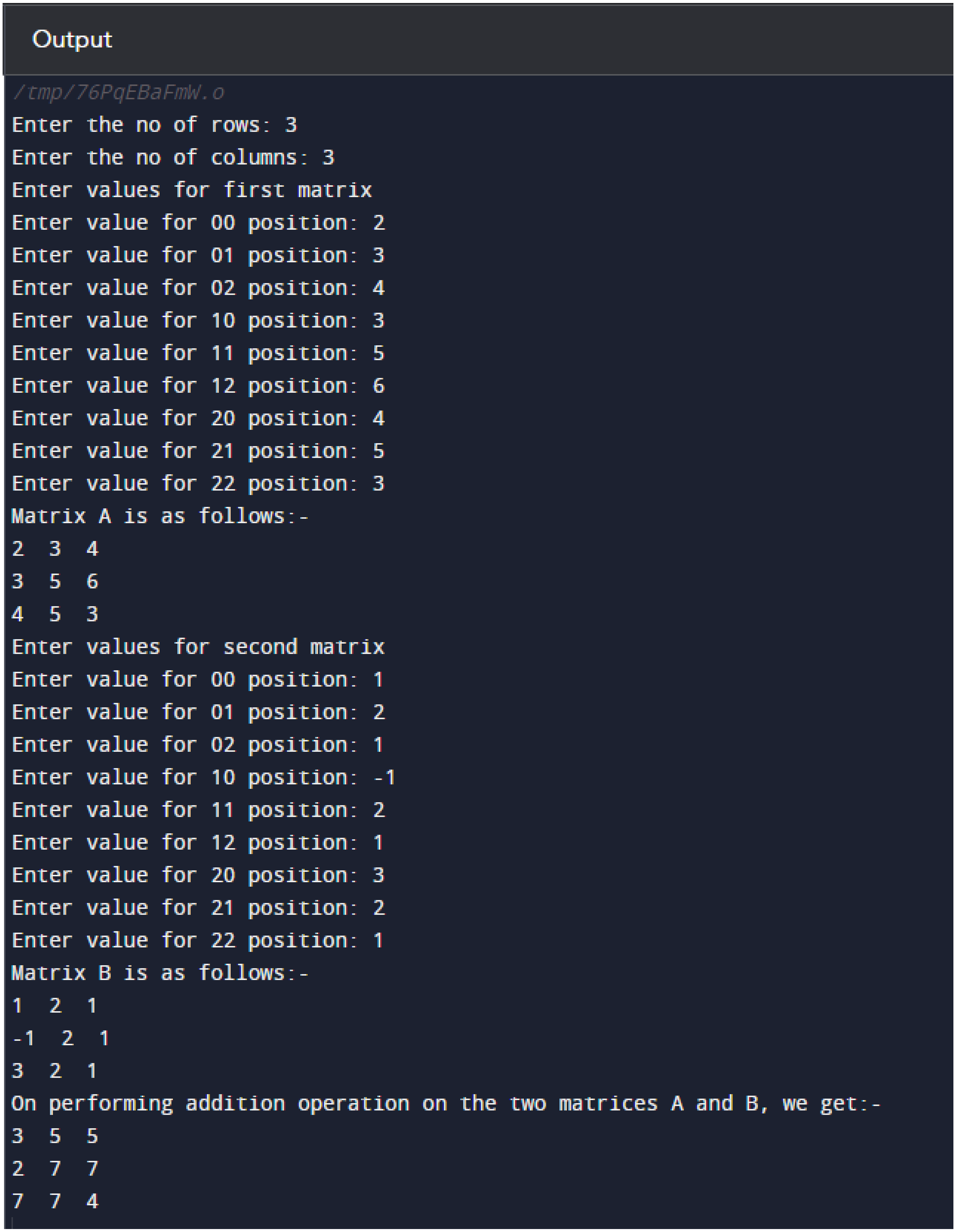
```c
// Program in C to use a three-dimensional array for implementing matrix
addition
#include <stdio.h>
int main()
{
    int r,c;
    printf("Enter the no of rows: ");
    scanf("%d", &r);
    printf("Enter the no of columns: ");
    scanf("%d", &c);
    int a[r][c], b[r][c];

    printf("Enter values for first matrix\n");
    for(int i=0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("Enter value for %d%d position: ", i,j);
            scanf("%d", &a[i][j]);
        }
    }
    printf("Matrix A is as follows:-\n");
    for(int i=0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("%d  ", a[i][j]);
        }
        printf("\n");
    }

    printf("Enter values for second matrix\n");
    for(int i=0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("Enter value for %d%d position: ", i,j);
            scanf("%d", &b[i][j]);
        }
    }
    printf("Matrix B is as follows:-\n");
    for(int i=0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("%d  ", b[i][j]);
        }
        printf("\n");
    }

    printf("On performing addition operation on the two matrices A and B, we get:- \n");
    for(int i=0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("%d  ", a[i][j]+b[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT SCREEN 2:-



```
/tmp/76PqEBaFmW.o
Enter the no of rows: 3
Enter the no of columns: 3
Enter values for first matrix
Enter value for 00 position: 2
Enter value for 01 position: 3
Enter value for 02 position: 4
Enter value for 10 position: 3
Enter value for 11 position: 5
Enter value for 12 position: 6
Enter value for 20 position: 4
Enter value for 21 position: 5
Enter value for 22 position: 3
Matrix A is as follows:-
2  3  4
3  5  6
4  5  3
Enter values for second matrix
Enter value for 00 position: 1
Enter value for 01 position: 2
Enter value for 02 position: 1
Enter value for 10 position: -1
Enter value for 11 position: 2
Enter value for 12 position: 1
Enter value for 20 position: 3
Enter value for 21 position: 2
Enter value for 22 position: 1
Matrix B is as follows:-
1  2  1
-1  2  1
3  2  1
On performing addition operation on the two matrices A and B, we get:-
3  5  5
2  7  7
7  7  4
```

CODE 3:- (Matrix Subtraction)

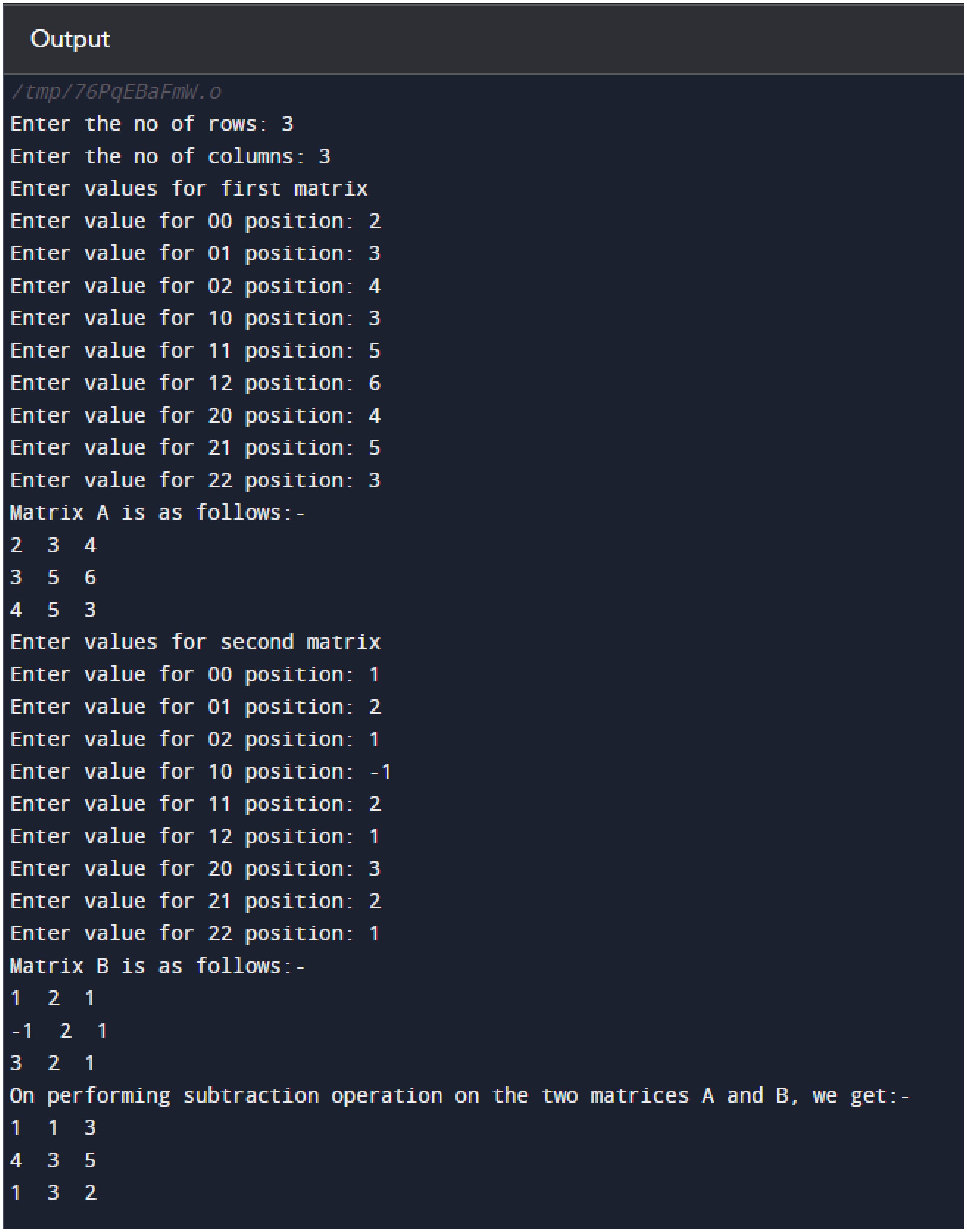
```c
// Program in C to use a three-dimensional array for implementing matrix subtraction
#include <stdio.h>
int main()
{
    int r,c;
    printf("Enter the no of rows: ");
    scanf("%d", &r);
    printf("Enter the no of columns: ");
    scanf("%d", &c);
    int a[r][c], b[r][c];

    printf("Enter values for first matrix\n");
    for(int i=0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("Enter value for %d%d position: ", i,j);
            scanf("%d", &a[i][j]);
        }
    }
    printf("Matrix A is as follows:-\n");
    for(int i=0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("%d  ", a[i][j]);
        }
        printf("\n");
    }

    printf("Enter values for second matrix\n");
    for(int i=0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("Enter value for %d%d position: ", i,j);
            scanf("%d", &b[i][j]);
        }
    }
    printf("Matrix B is as follows:-\n");
    for(int i=0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("%d  ", b[i][j]);
        }
        printf("\n");
    }

    printf("On performing subtraction operation on the two matrices A and
B, we get:- \n");
    for(int i=0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("%d  ", a[i][j]-b[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT SCREEN 3:-



```
/tmp/76PqEBaFmW.o
Enter the no of rows: 3
Enter the no of columns: 3
Enter values for first matrix
Enter value for 00 position: 2
Enter value for 01 position: 3
Enter value for 02 position: 4
Enter value for 10 position: 3
Enter value for 11 position: 5
Enter value for 12 position: 6
Enter value for 20 position: 4
Enter value for 21 position: 5
Enter value for 22 position: 3
Matrix A is as follows:-
2  3  4
3  5  6
4  5  3
Enter values for second matrix
Enter value for 00 position: 1
Enter value for 01 position: 2
Enter value for 02 position: 1
Enter value for 10 position: -1
Enter value for 11 position: 2
Enter value for 12 position: 1
Enter value for 20 position: 3
Enter value for 21 position: 2
Enter value for 22 position: 1
Matrix B is as follows:-
1  2  1
-1  2  1
3  2  1
On performing subtraction operation on the two matrices A and B, we get:-
1  1  3
4  3  5
1  3  2
```

CODE 4:- (Matrix Multiplication)

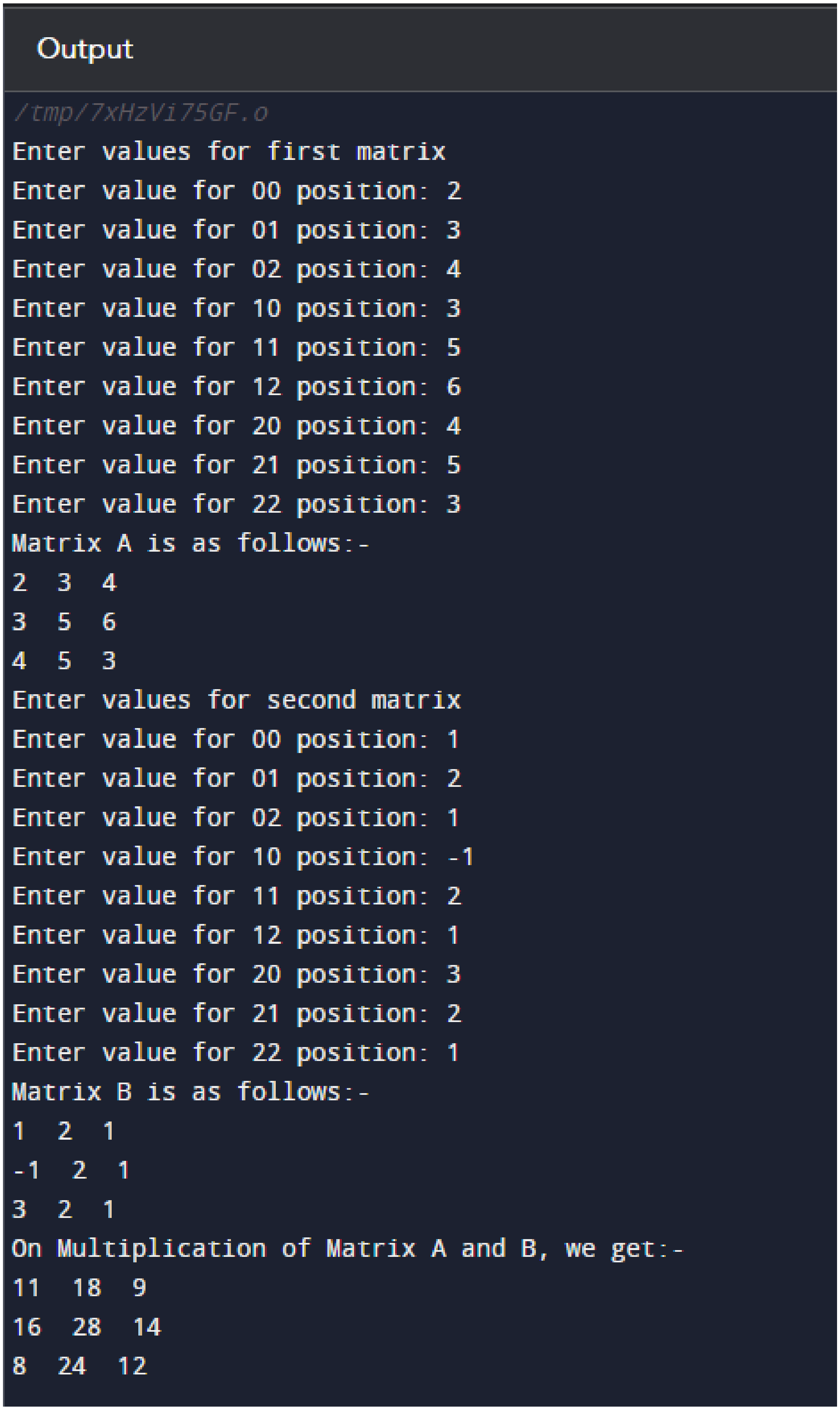
```
// Program in C to use a three-dimensional array for implementing matrix
     multiplication
#include <stdio.h>
int main()
{
    int a[3][3], b[3][3], c[3][3];

    printf("Enter values for first matrix\n");
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("Enter value for %d%d position: ", i,j);
            scanf("%d", &a[i][j]);
        }
    }
    printf("Matrix A is as follows:-\n");
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("%d  ", a[i][j]);
        }
        printf("\n");
    }

    printf("Enter values for second matrix\n");
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("Enter value for %d%d position: ", i,j);
            scanf("%d", &b[i][j]);
        }
    }
    printf("Matrix B is as follows:-\n");
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("%d  ", b[i][j]);
        }
        printf("\n");
    }

    // Matrix Multiplication and printing
    printf("On Multiplication of Matrix A and B, we get:-\n");
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            c[i][j]=0;
            for(int k=0;k<3;k++)
            {
                // You can use shorthand as well.
                // c[i][j]+= a[i][k]*b[i][k];
                c[i][j]= c[i][j] + (a[i][k]*b[k][j]);
            }
            printf("%d  ", c[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT SCREEN 4:-

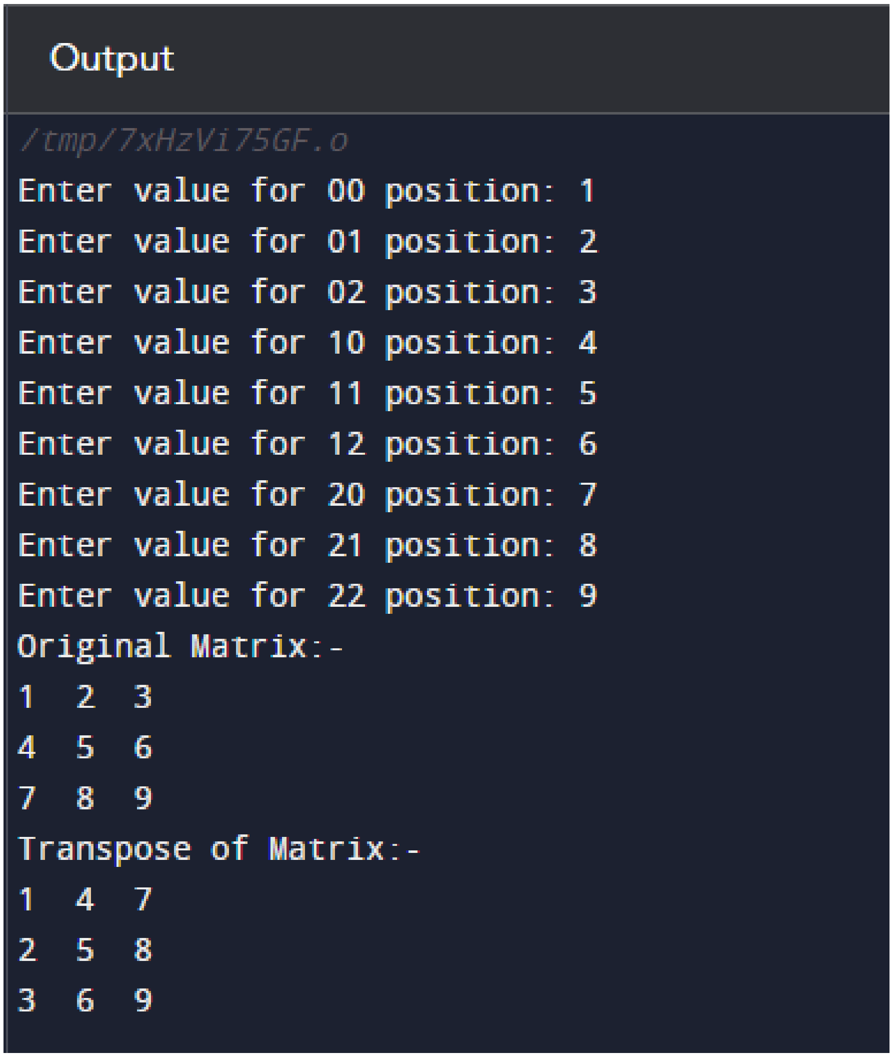


```
/tmp/7xHzVi75GF.o
Enter values for first matrix
Enter value for 00 position: 2
Enter value for 01 position: 3
Enter value for 02 position: 4
Enter value for 10 position: 3
Enter value for 11 position: 5
Enter value for 12 position: 6
Enter value for 20 position: 4
Enter value for 21 position: 5
Enter value for 22 position: 3
Matrix A is as follows:-
2  3  4
3  5  6
4  5  3
Enter values for second matrix
Enter value for 00 position: 1
Enter value for 01 position: 2
Enter value for 02 position: 1
Enter value for 10 position: -1
Enter value for 11 position: 2
Enter value for 12 position: 1
Enter value for 20 position: 3
Enter value for 21 position: 2
Enter value for 22 position: 1
Matrix B is as follows:-
1  2  1
-1  2  1
3  2  1
On Multiplication of Matrix A and B, we get:-
11  18  9
16  28  14
8  24  12
```

CODE 5:- (Transpose of Matrix)

```c
// Program in C to find the transpose of a matrix using three-dimensional
     arrays
#include <stdio.h>
int main()
{
    int m[3][3];
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("Enter value for %d%d position: ", i,j);
            scanf("%d", &m[i][j]);
        }
    }
    printf("Original Matrix:-\n");
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("%d  ", m[i][j]);
        }
        printf("\n");
    }
    printf("Transpose of Matrix:-\n");
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("%d  ", m[j][i]);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT SCREEN 5:-



```
/tmp/7xHzVi75GF.o
Enter value for 00 position: 1
Enter value for 01 position: 2
Enter value for 02 position: 3
Enter value for 10 position: 4
Enter value for 11 position: 5
Enter value for 12 position: 6
Enter value for 20 position: 7
Enter value for 21 position: 8
Enter value for 22 position: 9
Original Matrix:-
1  2  3
4  5  6
7  8  9
Transpose of Matrix:-
1  4  7
2  5  8
3  6  9
```

(4) AIM:-

To implement a program in C that merges two sorted arrays into a single sorted array.

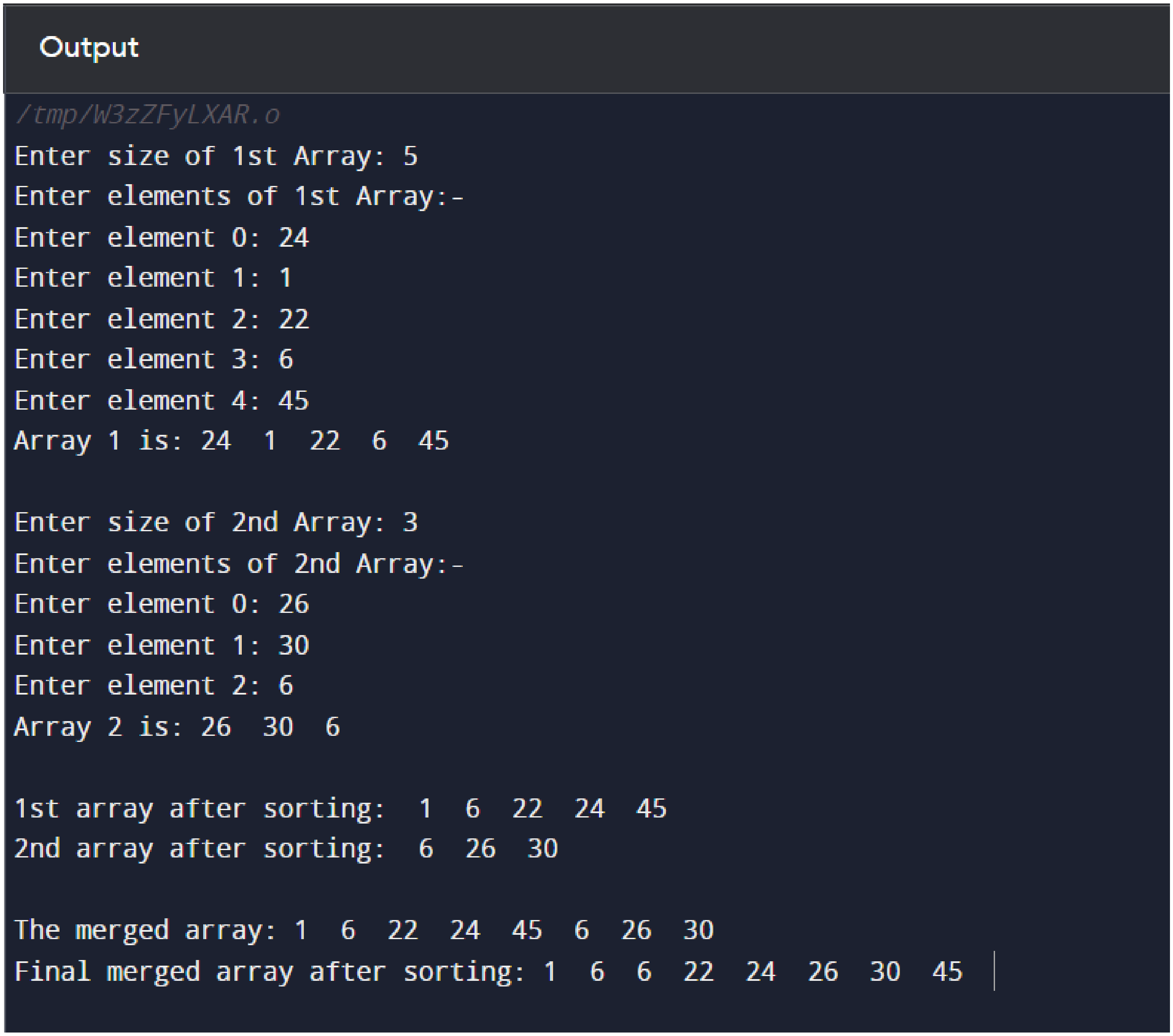
CODE:-

```
#include <stdio.h>
int main()
{
    int n1,n2, temp=0;
    printf("Enter size of 1st Array: ");
    scanf("%d", &n1);
    int a[n1];
    printf("Enter elements of 1st Array:-\n");
    for(int i=0; i<n1; i++)
    {
        printf("Enter element %d: ", i);
        scanf("%d", &a[i]);
    }
    printf("Array 1 is: ");
    for(int i=0; i<n1; i++)
    {
        printf("%d  ", a[i]);
    }
    printf("\n\nEnter size of 2nd Array: ");
    scanf("%d", &n2);
    int b[n2], c[n1+n2];
    printf("Enter elements of 2nd Array:-\n");
    for(int i=0; i<n2; i++)
    {
        printf("Enter element %d: ", i);
        scanf("%d", &b[i]);
    }
```

```c
printf("Array 2 is: ");
for(int i=0; i<n2; i++)
{
    printf("%d  ", b[i]);
}
// Sorting first array
for(int i=0; i<n1; i++)
{
    for(int j=i+1; j<n1; j++)
    {
        if(a[i] > a[j])
        {
            temp= a[i];
            a[i]= a[j];
            a[j]= temp;
        }
    }
}
printf("\n\n1st array after sorting:  ");
for(int i=0; i<n1; i++)
{
    printf("%d  ",a[i]);
}
// Sorting second array
for(int i=0; i<n2; i++)
{
    for(int j=i+1; j<n2; j++)
    {
        if(b[i] > b[j])
        {
            temp= b[i];
            b[i]= b[j];
            b[j]= temp;
        }
    }
}
printf("\n2nd array after sorting:  ");
for(int i=0; i<n2; i++)
{
    printf("%d  ",b[i]);
}
// Merging the two arrays
for(int i=0; i<n1; i++)
{
    c[i] = a[i];
}
for(int i = 0; i<n2; i++)
{
    c[i+n1] = b[i];
}
printf("\n\nThe merged array: ");
for(int i=0; i<(n1+n2); i++) //Printing the merged array
{
    printf("%d  ", c[i]);
}
// Sorting the merged array
printf("\nFinal merged array after sorting: ");
for(int i=0; i<(n1+n2); i++)
{
    for(int j=i+1; j<(n1+n2); j++)
    {
        if(c[i] > c[j])
        {
            temp= c[i];
            c[i]= c[j];
            c[j]= temp;
        }
    }
}
for(int i=0; i<(n1+n2); i++) //Printing final array
```

```
    {
        printf("%d  ",c[i]);
    }
    return 0;
}
```

OUTPUT SCREEN:-



Output

```
/tmp/W3zZFyLXAR.o
Enter size of 1st Array: 5
Enter elements of 1st Array:-
Enter element 0: 24
Enter element 1: 1
Enter element 2: 22
Enter element 3: 6
Enter element 4: 45
Array 1 is: 24  1  22  6  45

Enter size of 2nd Array: 3
Enter elements of 2nd Array:-
Enter element 0: 26
Enter element 1: 30
Enter element 2: 6
Array 2 is: 26  30  6

1st array after sorting:  1  6  22  24  45
2nd array after sorting:  6  26  30

The merged array: 1  6  22  24  45  6  26  30
Final merged array after sorting: 1  6  6  22  24  26  30  45
```

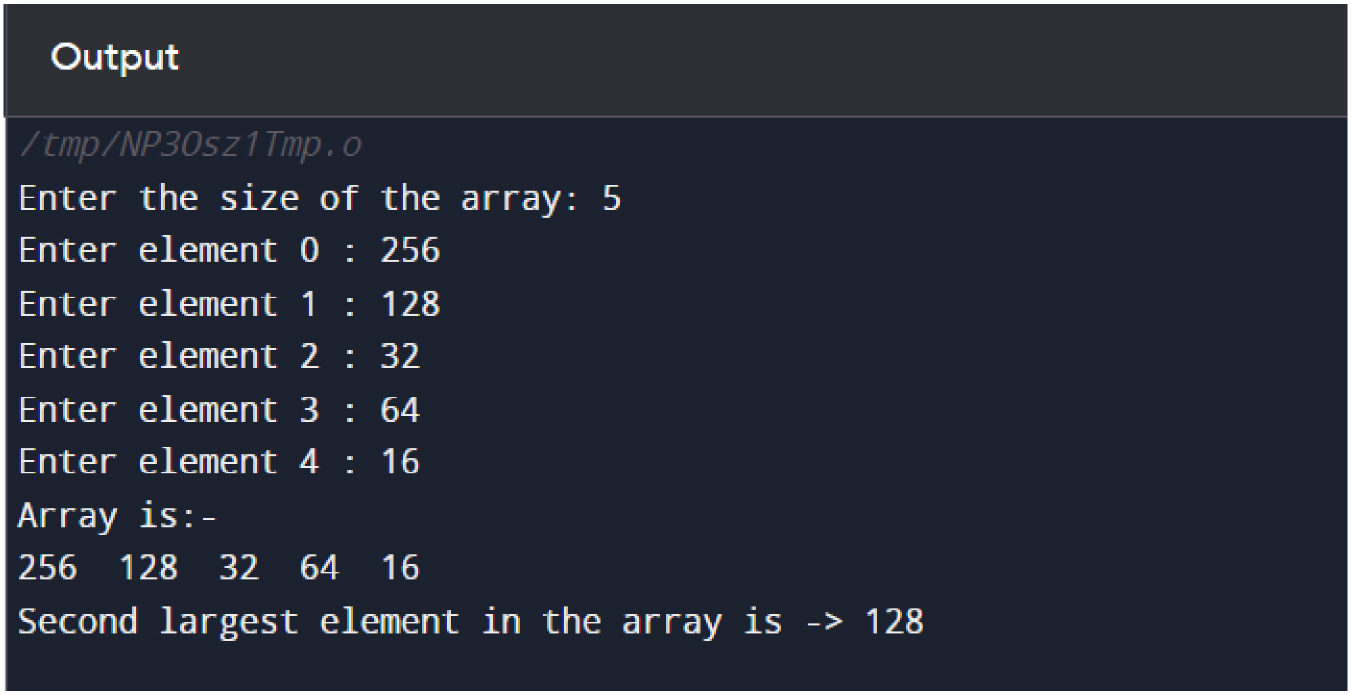
(5) AIM:-

To write a program to find the second largest element in an array.

CODE:-

```
// Program in C to find out and print the second largest element in the array out of all the user input
#include <stdio.h>
int main()
{
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int arr[n];
    for(int i=0; i<n; i++)
    {
        printf("Enter element %d : ",i);
        scanf("%d", &arr[i]);
    }
    printf("Array is:-\n");
    for(int i=0; i<n; i++)
    {
        printf("%d  ", arr[i]);
    }
    int lr= arr[0];
    int slr;
    int p=0;
    for(int i=0; i<n; i++)
    {
        if(arr[i]>lr)
        {
            lr= arr[i];
            p= i;
    }
    else
      slr= arr[i-1];
  }
  for(int i=0; i<n; i++)
  {
    if(i!=p)
    {
      if(arr[i]>slr)
        slr= arr[i];
    }
  }
  printf("\nSecond largest element in the array is -> %d", slr);
  return 0;
}
```

<u>OUTPUT SCREEN</u>:-



```
/tmp/NP3Osz1Tmp.o
Enter the size of the array: 5
Enter element 0 : 256
Enter element 1 : 128
Enter element 2 : 32
Enter element 3 : 64
Enter element 4 : 16
Array is:-
256  128  32  64  16
Second largest element in the array is -> 128
```

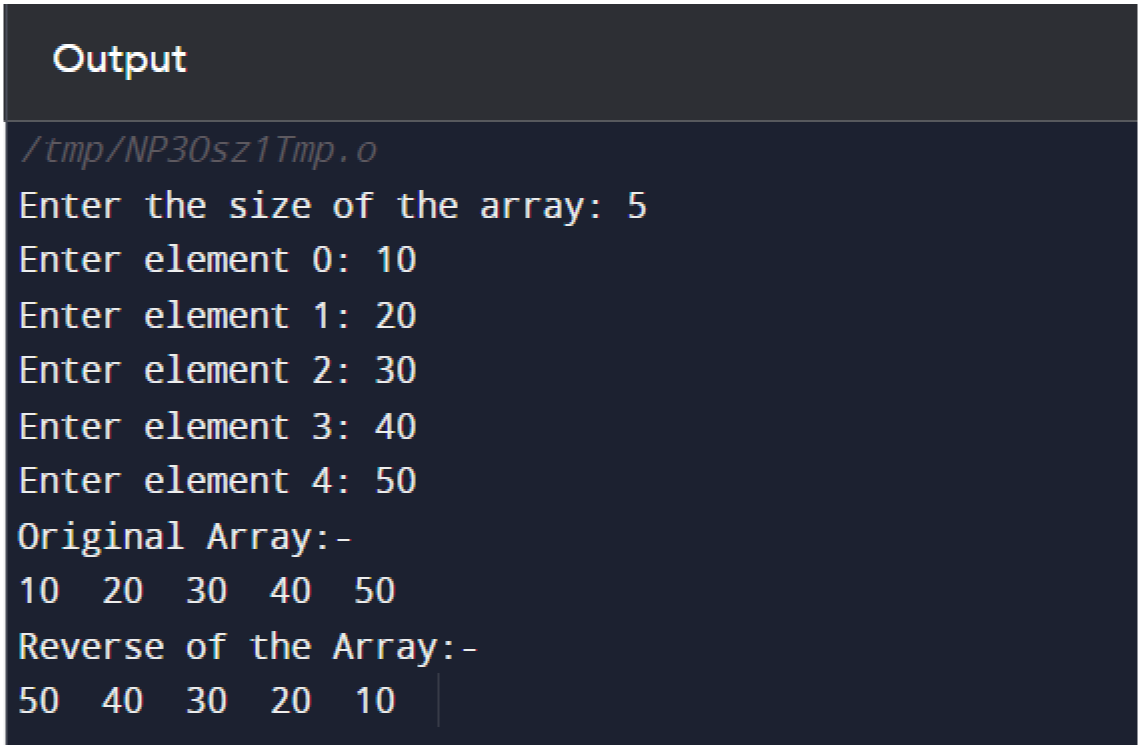
(6) AIM:-

To write a program to reverse the elements of an array.

CODE:-

```
// Program in C to reverse the elements of an array
#include <stdio.h>
int main()
{
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int arr[n];
    for(int i=0; i<n; i++)
    {
        printf("Enter element %d: ",i);
        scanf("%d", &arr[i]);
    }
    printf("Original Array:-\n");
    for(int i=0; i<n; i++)
    {
        printf("%d  ", arr[i]);
    }
    // Reversing the Array
    for(int i=0, j=n-1; i<j; i++, j--)
    {
        int temp= arr[i];
        arr[i]= arr[j];
        arr[j]= temp;
    }
    printf("\nReverse of the Array:-\n");
    for(int i=0; i<n; i++)
    {
        printf("%d  ", arr[i]);
    }
    return 0;
}
```

OUTPUT SCREEN:-



```
/tmp/NP30sz1Tmp.o
Enter the size of the array: 5
Enter element 0: 10
Enter element 1: 20
Enter element 2: 30
Enter element 3: 40
Enter element 4: 50
Original Array:-
10  20  30  40  50
Reverse of the Array:-
50  40  30  20  10
```