

## Lab Sheet 7

1. WAP to store 10 integer elements in an array and print it.

➤ **Code**

```
#include <stdio.h>

int main( ) {
    int arr[10];
    printf("Enter 10 integers: ");
    for (int i = 0; i < 10; i++) {
        scanf("%d", &arr[i]);
    }
    printf("The array is: ");
    for (int i = 0; i < 10; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

➤ **Output**

```
Enter 10 integers: 5 6 3 7 9 1 1 3 4 6
The array is: 5 6 3 7 9 1 1 3 4 6
```

2. WAP to calculate and print the sum and average of n elements of array

➤ **Code**

```
int main()
{
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d integers: ", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += arr[i];
    }
    printf("The sum is %d\n", sum);
    printf("The average is %f\n", (float)sum / n);
    return 0;
}
```

➤ **Output**

```
Enter the number of elements: 5
Enter 5 integers: 1 4 6 2 7
The sum is 20
The average is 4.000000
```

### 3. WAP to sort n elements of array in descending order.

#### ➤ Code

```
#include <stdio.h>

int main()
{
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d integers: ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    // Bubble sort

    for (int i = 0; i < n - 1; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (arr[i] < arr[j])
            {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
    printf("The sorted array is: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

#### ➤ Output

```
Enter the number of elements: 5
Enter 5 integers: 3 6 -2 9 0
The sorted array is: 9 6 3 0 -2
```

#### 4. WAP to count the frequency of the elements in an array.

##### ➤ Code

```
#include <stdio.h>

int main()
{
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d integers: ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }

    int count = 0, num;
    printf("Enter the number to count the frequency: ");
    scanf("%d", &num);
    for (int i = 0; i < n; i++)
    {
        if (arr[i] == num)
        {
            count++;
        }
    }
    printf("The frequency of %d in the given array is %d\n", num, count);

    return 0;
}
```

##### ➤ Output

```
Enter the number of elements: 10
Enter 10 integers: 1 5 2 6 3 6 7 1 2 9
Enter the number to count the frequency: 6
The frequency of 6 in the given array is 2
```

5. WAP to pass 1D array to the minmax() function. The function should calculate the maximum and minimum among the elements of the array. Print the maximum and minimum in the main() function.

➤ **Code**

```
#include <stdio.h>
void minmax(int arr[], int n, int *min, int *max)
{
    *min = arr[0];
    *max = arr[0];
    for (int i = 1; i < n; i++)
    {
        if (arr[i] < *min)
        {
            *min = arr[i];
        }
        if (arr[i] > *max)
        {
            *max = arr[i];
        }
    }
}
int main()
{
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d integers: ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    int min, max;
    minmax(arr, n, &min, &max);
    printf("The minimum is %d\n", min);
    printf("The maximum is %d\n", max);
    return 0;
}
```

➤ **Output**

```
Enter the number of elements: 10
Enter 10 integers: 1 3 5 0 -2 45 5 1 5 6
The minimum is -2
The maximum is 45
```

## 6. WAP to multiply matrix of different size if possible using 2D array

### ➤ Code

```
#include <stdio.h>
#define MAX_SIZE 10

void multiplyMatrices(int mat1[][MAX_SIZE], int rows1, int cols1, int mat2[][MAX_SIZE], int rows2, int cols2, int result[][MAX_SIZE])
{
    int i, j, k;
    // Multiply matrices
    for (i = 0; i < rows1; i++)
    {
        for (j = 0; j < cols2; j++)
        {
            result[i][j] = 0;
            for (k = 0; k < cols1; k++)
            {
                result[i][j] += mat1[i][k] * mat2[k][j];
            }
        }
    }
}

void readMatrix(int matrix[][MAX_SIZE], int rows, int cols)
{
    int i, j;
    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < cols; j++)
        {
            scanf("%d", &matrix[i][j]);
        }
    }
}

void printMatrix(int matrix[][MAX_SIZE], int rows, int cols)
{
    int i, j;
    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < cols; j++)
        {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
}
```

```
int main( )
{
    int mat1[MAX_SIZE][MAX_SIZE];
    int mat2[MAX_SIZE][MAX_SIZE];
    int result[MAX_SIZE][MAX_SIZE];
    int rows1, cols1, rows2, cols2;
    int i, j;

    printf("Enter the dimension of first matrix: ");
    scanf("%d %d", &rows1, &cols1);
    printf("Enter the dimension of second matrix: ");
    scanf("%d %d", &rows2, &cols2);

    // Check if matrices can be multiplied
    if (cols1 != rows2)
    {
        printf("Matrices cannot be multiplied.\n");
        return 0;
    }
    else
    {
        printf("Enter elements of second matrix:\n");
        readMatrix(mat2, rows2, cols2);
        printf("Enter elements of first matrix:\n");
        readMatrix(mat1, rows1, cols1);
        multiplyMatrices(mat1, rows1, cols1, mat2, rows2, cols2, result);
        printf("Product of the matrices:\n");
        printMatrix(result, rows1, cols2);
        return 0;
    }
}
```

### ➤ Output

```
Enter the dimension of first matrix: 2 3
Enter the dimension of second matrix: 3 2
Enter elements of second matrix:
1 2 3
4 5 6
Enter elements of first matrix:
10 11
20 21
30 31
Product of the matrices:
140 146
320 335
```

## 7. WAP to find the transpose of the matrix

### ➤ Code

```
#include <stdio.h>
#define MAX_SIZE 10

void transposeMatrix(int matrix[][MAX_SIZE], int
rows, int cols, int transpose[][MAX_SIZE])
{
    int i, j;

    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < cols; j++)
        {
            transpose[j][i] = matrix[i][j];
        }
    }
}

void printMatrix(int matrix[][MAX_SIZE], int rows,
int cols)
{
    int i, j;
    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < cols; j++)
        {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
}

int main()
{
    int matrix[MAX_SIZE][MAX_SIZE];
    int transpose[MAX_SIZE][MAX_SIZE];
    int rows, cols;
```

```
int i, j;

printf("Enter the dimensions of the matrix: ");
scanf("%d %d", &rows, &cols);

printf("Enter the elements of the matrix:\n");
for (i = 0; i < rows; i++)
{
    for (j = 0; j < cols; j++)
    {
        scanf("%d", &matrix[i][j]);
    }
}
transposeMatrix(matrix, rows, cols, transpose);
printf("Original matrix:\n");
printMatrix(matrix, rows, cols);
printf("Transpose matrix:\n");
printMatrix(transpose, cols, rows);
return 0;
}
```

### ➤ Output

```
Enter the elements of the matrix:
1 2 3
4 5 6
Original matrix:
1 2 3
4 5 6
Transpose matrix:
1 4
2 5
3 6
```

## 8. WAP to find the sum of the rows and column of the matrix.

### ➤ Code

```
#include <stdio.h>
#define MAX_SIZE 10

void calculateRowSum(int matrix[][MAX_SIZE], int
rows, int cols, int rowSum[])
{
    int i, j;

    for (i = 0; i < rows; i++)
    {
        rowSum[i] = 0;
        for (j = 0; j < cols; j++)
        {
            rowSum[i] += matrix[i][j];
        }
    }
}

void calculateColumnSum(int matrix[][MAX_SIZE],
int rows, int cols, int colSum[])
{
    int i, j;

    for (j = 0; j < cols; j++)
    {
        colSum[j] = 0;
        for (i = 0; i < rows; i++)
        {
            colSum[j] += matrix[i][j];
        }
    }
}

void printArray(int arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
```

```
int main()
{
    int matrix[MAX_SIZE][MAX_SIZE];
    int rowSum[MAX_SIZE];
    int colSum[MAX_SIZE];
    int rows, cols;
    int i, j;

    printf("Enter the number of rows and columns of
the matrix: ");
    scanf("%d %d", &rows, &cols);

    printf("Enter the elements of the matrix:\n");
    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < cols; j++)
        {
            scanf("%d", &matrix[i][j]);
        }
    }

    calculateRowSum(matrix, rows, cols, rowSum);
    calculateColumnSum(matrix, rows, cols, colSum);

    printf("Row sums: ");
    printArray(rowSum, rows);

    printf("Column sums: ");
    printArray(colSum, cols);

    return 0;
}
```

### ➤ Output

```
Enter the elements of the matrix:
1 2 3 4
5 6 7 8
Row sums: 10 26
Column sums: 6 8 10 12
```