

**EXPERIMENT NO. 5**

**Ques 1 :-** Write a program to implement the following scenarios. Take all the input from user, nothing should be imagine or hard coded.

1. - Transpose of a matrix.
2. - Check if a matrix is symmetrical or not.

```
#include <stdio.h>
```

```
int main() {
```

```
    int n, m;
```

```
    printf("Enter the number of rows: ");
```

```
    scanf("%d", &n);
```

```
    printf("Enter the number of columns: ");
```

```
    scanf("%d", &m);
```

```
    if (n != m) {
```

```
        printf("A non-square matrix cannot be symmetric.\n");
```

```
        return 0;
```

```
    }
```

```
    int matrix[n][m], transpose[m][n];
```

```
    printf("Enter the matrix elements:\n");
```

```
for (int i = 1; i <= n; i++) {  
    for (int j = 1; j <= m; j++) {  
        printf("Enter element at position (%d, %d): ", i , j );  
        scanf("%d", &matrix[i][j]);  
    }  
}
```

```
int Symm = 1;  
for (int i = 1; i <=n; i++) {  
    for (int j = 1; j <= m; j++) {  
        if (matrix[i][j] != matrix[j][i]) {  
            Symm = 0;  
            break;  
        }  
    }  
    if (!Symm) {  
        break;  
    }  
}
```

```
printf("\nOriginal Matrix:\n");
```

```
for (int i = 1; i <= n; i++) {  
    for (int j = 1; j <= m; j++) {  
        printf("%d\t", matrix[i][j]);  
    }  
    printf("\n");  
}  
  
printf("\nTransposed Matrix:\n");  
for (int i = 1; i <= m; i++) {  
    for (int j = 1; j <= n; j++) {  
        printf("%d\t", matrix[j][i]);  
    }  
    printf("\n");  
}  
  
if (Symm) {  
    printf("\nThe matrix is symmetric.\n");  
} else {  
    printf("\nThe matrix is not symmetric.\n");  
}  
  
return 0;
```

```
}
```

### Output of the Program :-

```
Enter the number of rows: 3
Enter the number of columns: 3
Enter the matrix elements:
Enter element at position (1, 1): 1
Enter element at position (1, 2): 2
Enter element at position (1, 3): 3
Enter element at position (2, 1): 4
Enter element at position (2, 2): 5
Enter element at position (2, 3): 6
Enter element at position (3, 1): 7
Enter element at position (3, 2): 8
Enter element at position (3, 3): 9

Original Matrix:
1   2   3
4   5   6
7   8   9

Transposed Matrix:
1   4   7
2   5   8
3   6   9

The matrix is not symmetric.
PS E:\Data Structure and Algorithm In C\Experiment 5>
```

**Ques 2 :- Write a program to merge to array and append them in the following order.**

- 1. – Add the first array to the end of the another one.**
- 2. – Add second array to the end of the first one.**
- 3. – Merge the array and sort them.**

```
#include <stdio.h>
```

```
int main() {
```

```
    int size1, size2;
```

```
    printf("Enter the size of the first array: ");
```

```
    scanf("%d", &size1);
```

```
    int arr1[size1];
```

```
    printf("Enter elements of the first array:\n");
```

```
    for (int i = 0; i < size1; i++) {
```

```
        printf("Enter element %d: ", i + 1);
```

```
        scanf("%d", &arr1[i]);
```

```
    }
```

```
    printf("Enter the size of the second array: ");
```

```
    scanf("%d", &size2);
```

```
    int arr2[size2];
```

```
    printf("Enter elements of the second array:\n");
```

```
    for (int i = 0; i < size2; i++) {
```

```
        printf("Enter element %d: ", i + 1);
```

```
        scanf("%d", &arr2[i]);
```

```
}

int totalSize = size1 + size2;
int result[totalSize];

for (int i = 0; i < size2; i++) {
    result[i] = arr2[i];
}

for (int i = 0; i < size1; i++) {
    result[size2 + i] = arr1[i];
}

printf("\nMerged Array:\n");
for (int i = 0; i < totalSize; i++) {
    printf("%d ", result[i]);
}

return 0;
}
```

**Output of the Program :-**

```
Enter the size of the first array: 3
Enter elements of the first array:
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter the size of the second array: 4
Enter elements of the second array:
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 4
```

```
Merged Array:
```

```
1 2 3 4 1 2 3
```

```
#include <stdio.h>
```

```
int main() {
```

```
    int size1, size2;
```

```
    printf("Enter the size of the first array: ");
```

```
    scanf("%d", &size1);
```

```
    int arr1[size1];
```

```
    printf("Enter elements of the first array:\n");
```

```
    for (int i = 0; i < size1; i++) {
```

```
        printf("Enter element %d: ", i + 1);
```

```
        scanf("%d", &arr1[i]);
```

```
}
```

```
printf("Enter the size of the second array: ");
```

```
scanf("%d", &size2);
```

```
int arr2[size2];
```

```
printf("Enter elements of the second array:\n");
```

```
for (int i = 0; i < size2; i++) {
```

```
    printf("Enter element %d: ", i + 1);
```

```
    scanf("%d", &arr2[i]);
```

```
}
```

```
size1 += size2;
```

```
for (int i = 0; i < size2; i++) {
```

```
    arr1[size1 - size2 + i] = arr2[i];
```

```
}
```

```
printf("\nMerged Array:\n");
```

```
for (int i = 0; i < size1; i++) {
```

```
    printf("%d ", arr1[i]);
```

```
}
```



```
    return 0;
}
```

### Output of the Program :-

```
Enter the size of the first array: 4
Enter elements of the first array:
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 4
Enter the size of the second array: 4
Enter elements of the second array:
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 10

Merged Array:
1 2 3 4 1 2 3 10
PS E:\Data Structure and Algorithm In C\Experiment 5> |
```

```
#include<stdio.h>

int main()
{
    int n;

    printf("enter the value of n : ");
    scanf("%d", &n);

    int arr[n];

    printf("enter the values of array \n: ");
```

```
for (int i = 0; i < n; i++)  
{  
  
    scanf("%d", &arr[i]);  
}  
for (int i = 0; i < n; i++)  
{  
  
    for (int j =0; j < n-1; j++)  
    {  
        if (arr[j] > arr[j + 1])  
        {  
            int temp = arr[j];  
            arr[j] = arr[j + 1];  
            arr[j + 1] = temp;  
        }  
    }  
}
```

```
for(int i=0;i<n;i++){
```

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

### Output of the Program :-

```
enter the value of n : 5  
enter the values of array  
: 10  
44  
66  
77  
88  
10 44 66 77 88  
PS E:\Data Structure and Algorithm In C\Experiment 5> |
```

**Ques 3 :- Write a program using pointers to find the smallest number in an array using pointer?**

```
#include<stdio.h>  
  
int findSmallest(int *arr,int size)  
{  
    int smallest_number=*arr;  
    for(int i=0;i<size;i++)
```

```
{
    if(smallest_number > arr[i])
    {
        smallest_number = arr[i];
    }
}
return smallest_number;
}
int main()
{
    int n;
    printf("Enter the size of the array :\n");
    scanf("%d",&n);
    int arr[n];
    printf("Enter the values :\n");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int smallest = findSmallest(&arr[0],n);
    printf("Smallest Number of Array is %d",smallest);
```

```
}
```

### Output of the Program :-

```
Enter the size of the array :  
5  
Enter the values :  
10  
55  
33  
22  
89  
Smallest Number of Array is 10  
PS E:\Data Structure and Algorithm In C\Experiment 5> []
```

**Ques 4 :- Write a program which performs following task.**

1. – Initialize an integer array of 10 elements in main().
2. – Pass the entire array to the function modify.
3. – In modify() multiply (You can use division, multiplication or subtraction) each element of array by 3.
4. – Return the control to main() and print the new array elements in main().

```
#include<stdio.h>
```

```
void modify(int arr[], int size)
```

```
{
```

```
    for(int i=0;i<10;i++)
```

```
    {
```

```
        arr[i] = arr[i]*3;
    }
}

int main()
{
    int arr[10];
    printf("Enter the value in array :\n");
    for(int i=0;i<10;i++)
    {
        scanf("%d",&arr[i]);
    }
    modify(arr,10);
    printf("Modified array elements :\n");
    for(int i=0;i<10;i++)
    {
        printf("%d ",arr[i]);
    }
    return 0;
}
```

**Output of the Program :-**

```
Enter the value in array :
```

```
10
```

```
20
```

```
30
```

```
40
```

```
50
```

```
60
```

```
70
```

```
80
```

```
90
```

```
100
```

```
Modified array elements :
```

```
30 60 90 120 150 180 210 240 270 300
```

```
PS E:\Data Structure and Algorithm In C\Experiment 5> |
```