Embedded Beginner Week 7

1. CodeForWin
    - Solve Introduction to Arrays Problems: 1, 3, 5, 8, 9, 13, 15, 16, 19.

P1:

```
#include <stdio.h>
#define MAX_SIZE 1000 // Maximum array size

int main()
{
    int arr[MAX_SIZE];
    int i, N;


    printf("Enter size of array: ");
    scanf("%d", &N);


    printf("Enter %d elements in the array : ", N);
    for(i=0; i<N; i++)
    {
        scanf("%d", &arr[i]);
    }


for(i=0; i<N; i++)
    {
        printf("%d, ", arr[i]);
    }
```

```
    return 0;
}
```

P3:

```
#include <stdio.h>

int main()
{
    int arr[100];
    int i, n, sum=0;


    printf("Enter size of the array: ");
    scanf("%d", &n);


    printf("Enter %d elements in the array: ", n);
    for(i=0; i<n; i++)
    {
        scanf("%d", &arr[i]);


        sum += arr[i];
    }

    printf("Sum of all elements of array = %d", sum);

    return 0;
}
```

P5:

```c
#include <stdio.h>


int main()
 {
    int arr[100], i, largest, secondLargest, n;

    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);

    printf("Enter the elements: ");
    scanf("%d", &arr[0]);


    largest = arr[0];                    //  // largest = secon
    secondLargest = arr[0];


    for(i = 1; i < n; i++)
    {
        scanf("%d", &arr[i]);

        if(arr[i] > largest)
        {
            secondLargest = largest;
            largest = arr[i];
        }

        else if(arr[i] > secondLargest && arr[i] != largest)
        {
            secondLargest = arr[i];
        }
    }

    printf("The largest element is: %d\n", largest);
```

```
    printf("The second largest element is: %d\n", secondLargest);

    return 0;
}
```

P8:

```
#include <stdio.h>


int main()
{
    int arr1[100], arr2[100];
    int i, N;

    printf("Enter the N of the array : ");
    scanf("%d", &N);

    printf("Enter elements of arr1 array : ");
    for(i=0; i<N; i++)
    {
        scanf("%d", &arr1[i]);
    }


    for(i=0; i<N; i++)
    {
        arr2[i] = arr1[i];
    }


    printf("\nElements of arr2 array are : ");
    for(i=0; i<N; i++)
    {
```

```
        printf("%d,", arr2[i]);
    }

    return 0;
}
```

P9:

```
#include <stdio.h>

int main()
{
    int arr[100];
    int i, N, num, pos;


    printf("Enter size of the array : ");
    scanf("%d", &N);

    printf("Enter elements in array : ");
    for(i=0; i<N; i++)
    {
        scanf("%d", &arr[i]);
    }


    printf("Enter element to insert : ");
    scanf("%d", &num);
    printf("Enter the element position : ");
    scanf("%d", &pos);


    if(pos > N+1 || pos <= 0)
    {
```

```
        printf("Invalid position! Please enter position between
    }
    else
    {

        for(i=N; i>=pos; i--)
        {
            arr[i] = arr[i-1];
        }


        arr[pos-1] = num;
        N++;


        printf("Array elements after insertion : ");
        for(i=0; i<N; i++)
        {
            printf("%d\t", arr[i]);
        }
    }

    return 0;
}
```

P13:

```
#include <stdio.h>

int main()
{
    int arr[100], n, i, j, count;

    printf("Enter the number of elements in the array: ");
```

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

    printf("Enter the elements of the array: ");

    for(i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }

    count = 0;
    for(i = 0; i < n; i++)
    {
        for(j = i+1; j < n; j++)
        {
            if(arr[i] == arr[j])
            {
                count++;
                break;
            }
        }
    }
    printf("The number of duplicate elements in the array is: %(

    return 0;
}
```

P5:

```
#include <stdio.h>

int main()
{
    int arr1[100], arr2[100], mergeArray[100* 2];   // note size
    int size1, size2, mergeSize, index1, index2, mergeIndex;
    int i;
```

```
printf("Enter the size of first array : ");
scanf("%d", &size1);

printf("\nEnter the size of second array : ");
scanf("%d", &size2);

printf("Enter elements in first array : ");
for(i=0; i<size1; i++)
{
    scanf("%d", &arr1[i]);
}


printf("Enter elements in second array : ");
for(i=0; i<size2; i++)
{
    scanf("%d", &arr2[i]);
}


mergeSize = size1 + size2;



index1 = 0;
index2 = 0;

for(mergeIndex=0; mergeIndex < mergeSize; mergeIndex++)
{

    if(index1 >= size1 || index2 >= size2)
    {
        break;
    }
```

```
        if(arr1[index1] < arr2[index2])
        {
            mergeArray[mergeIndex] = arr1[index1];
            index1++;
        }
        else
        {
            mergeArray[mergeIndex] = arr2[index2];
            index2++;
        }
    }

    while(index1 < size1)
    {
        mergeArray[mergeIndex] = arr1[index1];
        mergeIndex++;
        index1++;
    }
    while(index2 < size2)
    {
        mergeArray[mergeIndex] = arr2[index2];
        mergeIndex++;
        index2++;
    }



    printf("\nArray merged in ascending order : ");
    for(i=0; i<mergeSize; i++)
    {
        printf("%d\t", mergeArray[i]);
    }
```

```
    return 0;
}
```

P16:

```
#include <stdio.h>


int main()
{
    int arr[100];
    int size, i, arrIndex, revIndex;
    int temp;


    printf("Enter size of the array: ");
    scanf("%d", &size);


    printf("Enter elements in array: ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }

    revIndex = 0;
    arrIndex = size - 1;

    while(revIndex < arrIndex)
    {

        temp = arr[revIndex];
        arr[revIndex] = arr[arrIndex];
        arr[arrIndex] = temp;
```

```
        revIndex++;
        arrIndex--;
    }


    printf("\nReversed array : ");

    for(i=0; i<size; i++)
    {
        printf("%d\t", arr[i]);
    }

    return 0;
}
```

P19:

```
#include <stdio.h>

int main()
{
    int arr[100], n, i, j, temp, AorD;

    printf("Enter size of array: ");
    scanf("%d", &n);

    printf("Enter elements in array: ");
    for(i=0; i<n; i++)
    {
        scanf("%d", &arr[i]);
    }

    printf("write (1) for ascending order or (2) for decending o
```

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

    if(AorD == 1)
    {
        for(i=0; i<n; i++)
        {
            for(j=i+1; j<n; j++)
            {
                if(arr[i] > arr[j])
                {
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }

        printf("Sorted array in ascending order: ");
        for(i=0; i<n; i++)
        {
            printf("%d ", arr[i]);
        }
    }
    else if(AorD == 2)
    {
        for(i=0; i<n; i++)
        {
            for(j=i+1; j<n; j++)
            {
                if(arr[i] < arr[j])
                {
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
```

```
        }

        printf("Sorted array in descending order: ");
        for(i=0; i<n; i++)
        {
            printf("%d ", arr[i]);
        }
    }
    else
    {
        printf("Error number not valid");
    }

    return 0;
}
```

- Solve Multi-Dimensional Arrays Problems: 24, 27, 30, 37, 40.

P24:

```
#include <stdio.h>

#define SIZE 3 // Size of the matrix

int main()
{
    int A[SIZE][SIZE];
    int B[SIZE][SIZE];
    int C[SIZE][SIZE];

    int row, col;

    printf("Enter elements in matrix A of size 3x3: \n");
```

```
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            scanf("%d", &A[row][col]);
        }
    }


    printf("\nEnter elements in matrix B of size 3x3: \n");
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            scanf("%d", &B[row][col]);
        }
    }


    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {

            C[row][col] = A[row][col] - B[row][col];
        }
    }


    printf("\nDifference of two matrices A-B = \n");
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            printf("%d ", C[row][col]);
        }
```

```
        printf("\n");
    }

    return 0;
}
```

P27:

```
#include <stdio.h>

#define SIZE 3 // Matrix size

int main()
{
    int A[SIZE][SIZE];
    int B[SIZE][SIZE];

    int row, col, isEqual;

    printf("Enter elements in matrix A of size %dx%d: \n", SIZE,
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            scanf("%d", &A[row][col]);
        }
    }


    printf("\nEnter elements in matrix B of size %dx%d: \n");
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
```

```
            scanf("%d", &B[row][col]);
        }
    }


    isEqual = 1;

    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {

            if(A[row][col] != B[row][col])
            {
                isEqual = 0;
                break;
            }
        }
    }


    if(isEqual == 1)
    {
        printf("\nMatrix A is equal to Matrix B");
    }
    else
    {
        printf("\nMatrix A is not equal to Matrix B");
    }

    return 0;
}
```

P30:

```c
#include <stdio.h>

#define SIZE 3 // Matrix size

int main()
{
    int A[SIZE][SIZE];
    int row, col, sum = 0;


    printf("Enter elements in matrix of size %dx%d: \n", SIZE, S
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            scanf("%d", &A[row][col]);
        }
    }


    for(row=0; row<SIZE; row++)
    {
        sum = 0;
        for(col=0; col<SIZE; col++)
        {
            sum += A[row][col];
        }

        printf("Sum of elements of Row %d = %d\n", row+1, sum);
    }


    for(row=0; row<SIZE; row++)
    {
        sum = 0;
```

```
        for(col=0; col<SIZE; col++)
        {
            sum += A[col][row];
        }

        printf("Sum of elements of Column %d = %d\n", row+1, sum
    }

    return 0;
}
```

P37:

```
#include <stdio.h>

int main() {
    int SIZE;
    int A[100][100];
    int row, col;
    long det;
    int a, b, c, d, e, f, g, h, i;

    printf("Enter size of matrix (2 or 3): \n");

    scanf("%d", &SIZE);

    printf("Enter elements in matrix: \n");

    for(row=0; row<SIZE; row++) {
        for(col=0; col<SIZE; col++) {
            scanf("%d", &A[row][col]);
        }
    }
```

```
    if (SIZE == 2) {
        det = (A[0][0] * A[1][1]) - (A[0][1] * A[1][0]);

        printf("Determinant of matrix A = %ld", det);
    }

    else if (SIZE == 3) {
        a = A[0][0];
        b = A[0][1];
        c = A[0][2];
        d = A[1][0];
        e = A[1][1];
        f = A[1][2];
        g = A[2][0];
        h = A[2][1];
        i = A[2][2];

        det = (a*(e*i - f*h)) - (b*(d*i - f*g)) + (c*(d*h - e*g

        printf("Determinant of matrix A = %ld", det);
    }
    else {
        printf("Option not valid");
    }

    return 0;
}
```

P40:

```
#include <stdio.h>
#define SIZE 3

int main()
```

```
{
    int A[SIZE][SIZE];
    int B[SIZE][SIZE];
    int row, col, isSymmetric;

    printf("Enter elements in matrix of size 3x3: \n");
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            scanf("%d", &A[row][col]);
        }
    }


    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {

            B[row][col] = A[col][row];
        }
    }



    isSymmetric = 1;
    for(row=0; row<SIZE && isSymmetric; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            /* If matrix A is not equal to its transpose */
            if(A[row][col] != B[row][col])
            {
                isSymmetric = 0;
                break;
```

```
            }
        }
    }

    if(isSymmetric == 1)
    {
        printf("\nThe given matrix is Symmetric matrix: \n");

        for(row=0; row<SIZE; row++)
        {
            for(col=0; col<SIZE; col++)
            {
                printf("%d ", A[row][col]);
            }

            printf("\n");
        }
    }
    else
    {
        printf("\nThe given matrix is not Symmetric matrix.");
    }

    return 0;
}
```

1. HackerRank
    - Solve Problems: 1D Arrays in C

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

```c
#include <stdlib.h>

int main() {
    int n;
    scanf("%d", &n);

    int arr[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("%d", sum);
    return 0;
}
```