Problems

- Find sum of all array elements using recursion.

```
#include <stdio.h>

int sumArray(int arr[], int n) {
    if (n == 0) return 0;
    return arr[n - 1] + sumArray(arr, n - 1);
}

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int sum = sumArray(arr, 5);
    printf("Sum: %d\n", sum);
    return 0;
}
```

- Create an array ‘a1’ with ‘n’ elements. Insert an element in ith position of ‘a1’ and also delete an element from jth position of ‘a1’.

```
#include <stdio.h>

int sumArray(int arr[], int n) {
    if (n == 0) return 0;
    return arr[n - 1] + sumArray(arr, n - 1);
}

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int sum = sumArray(arr, 5);
    printf("Sum: %d\n", sum);
    return 0;
}
```

- Convert uppercase string to lowercase using for loop.

```
#include <stdio.h>

void toLowercase(char str[]) {
    for (int i = 0; str[i]; i++) {
        if (str[i] >= 'A' && str[i] <= 'Z') str[i] += 32;
    }
}

int main() {
    char str[] = "Hello World!";
    toLowercase(str);
    printf("%s\n", str);
    return 0;
}
```

- Find the sum of rows and columns of matrix of given order (row x column).

```
#include <stdio.h>

int main() {
    int r, c;
    scanf("%d %d", &r, &c);
    int matrix[r][c], rowSum[r], colSum[c];

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

```
        rowSum[i] = 0;
        for (int j = 0; j < c; j++) {
            scanf("%d", &matrix[i][j]);
            rowSum[i] += matrix[i][j];
            if (i == 0) colSum[j] = 0;
            colSum[j] += matrix[i][j];
        }
    }

    for (int i = 0; i < r; i++) printf("%d ", rowSum[i]);
    printf("\n");
    for (int j = 0; j < c; j++) printf("%d ", colSum[j]);
    printf("\n");

    return 0;
}
```

- Find the product of two matrices using pointers.

```
#include <stdio.h>

void multiplyMatrices(int (*a)[10], int (*b)[10], int (*result)[10], int r1, int c1, int c2) {
    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c2; j++) {
            result[i][j] = 0;
            for (int k = 0; k < c1; k++) {
                result[i][j] += a[i][k] * b[k][j];
            }
        }
    }
}

int main() {
    int a[10][10], b[10][10], result[10][10];
    int r1, c1, r2, c2;

    scanf("%d %d", &r1, &c1);
    scanf("%d %d", &r2, &c2);

    if (c1 != r2) return 0;

    for (int i = 0; i < r1; i++) for (int j = 0; j < c1; j++) scanf("%d", &a[i][j]);
    for (int i = 0; i < r2; i++) for (int j = 0; j < c2; j++) scanf("%d", &b[i][j]);

    multiplyMatrices(a, b, result, r1, c1, c2);

    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c2; j++) printf("%d ", result[i][j]);
        printf("\n");
    }

    return 0;
}
```

- Store ‘n’ numbers (integers or real) in an array. Conduct a linear search for a given number and report success or failure in the form of a suitable message.

```c
#include <stdio.h>

int linearSearch(int arr[], int n, int key) {
    for (int i = 0; i < n; i++) {
        if (arr[i] == key) return i;
    }
    return -1;
}

int main() {
    int n, key;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    int arr[n];

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

    printf("Enter the number to search: ");
    scanf("%d", &key);

    int result = linearSearch(arr, n, key);
    if (result != -1) {
        printf("Number found at index: %d\n", result);
    } else {
        printf("Number not found.\n");
    }
    return 0;
}
```

- Write a program to reverse an array.

```c
#include <stdio.h>

void reverseArray(int arr[], int n) {
    for (int i = 0; i < n / 2; i++) {
        int temp = arr[i];
        arr[i] = arr[n - 1 - i];
        arr[n - 1 - i] = temp;
    }
}

int main() {
    int n;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    int arr[n];

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

    reverseArray(arr, n);
    printf("Reversed array: ");
    for (int i = 0; i < n; i++) printf("%d ", arr[i]);
    printf("\n");
    return 0;
}
```

- Find the largest three distinct elements in an array: Input: arr[] = {10, 4, 3, 50, 23, 90} Output: 90, 50, 23

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

void findLargestThree(int arr[], int n) {
    int first = INT_MIN, second = INT_MIN, third = INT_MIN;

    for (int i = 0; i < n; i++) {
        if (arr[i] > first) {
            third = second;
            second = first;
            first = arr[i];
        } else if (arr[i] > second && arr[i] != first) {
            third = second;
            second = arr[i];
        } else if (arr[i] > third && arr[i] != second && arr[i] != first) {
            third = arr[i];
        }
    }

    if (third == INT_MIN) {
        printf("Not enough distinct elements.\n");
    } else {
        printf("Largest three distinct elements: %d, %d, %d\n", first, second, third);
    }
}

int main() {
    int arr[] = {10, 4, 3, 50, 23, 90};
    int n = sizeof(arr) / sizeof(arr[0]);
    findLargestThree(arr, n);
    return 0;
}
```

- Move all zeroes to end of array

```c
#include <stdio.h>

void moveZeroes(int arr[], int n) {
    int count = 0;

    for (int i = 0; i < n; i++) {
        if (arr[i] != 0) {
            arr[count++] = arr[i];
        }
    }
    while (count < n) arr[count++] = 0;
}

int main() {
    int arr[] = {0, 1, 0, 3, 12};
    int n = sizeof(arr) / sizeof(arr[0]);

    moveZeroes(arr, n);
    printf("Array after moving zeroes: ");
    for (int i = 0; i < n; i++) printf("%d ", arr[i]);
    printf("\n");
    return 0;
}
```

- Rearrange an array in maximum minimum form using Two Pointer Technique. Input: arr[] = {1, 2, 3, 4, 5, 6, 7} Output: arr[] = {7, 1, 6, 2, 5, 3, 4}.

```
#include <stdio.h>

void rearrange(int arr[], int n) {
    int temp[n];
    int start = 0, end = n - 1;

    for (int i = 0; i < n; i++) {
        if (i % 2 == 0) {
            temp[i] = arr[end--]; // Maximum element
        } else {
            temp[i] = arr[start++]; // Minimum element
        }
    }

    for (int i = 0; i < n; i++) {
        arr[i] = temp[i];
    }
}

int main() {
    int arr[] = {1, 2, 3, 4, 5, 6, 7};
    int n = sizeof(arr) / sizeof(arr[0]);

    rearrange(arr, n);
    printf("Rearranged array: ");
    for (int i = 0; i < n; i++) printf("%d ", arr[i]);
    printf("\n");
    return 0;
}
```

- Print all Distinct (Unique) Elements in given Array: Input: arr[] = {12, 10, 9, 45, 2, 10, 10, 45} Output: 12, 10, 9, 2

```
#include <stdio.h>

void printDistinct(int arr[], int n) {
    int found[n];
    int count = 0;

    for (int i = 0; i < n; i++) {
        int j;
        for (j = 0; j < count; j++) {
            if (arr[i] == found[j]) break;
        }
        if (j == count) {
            found[count++] = arr[i];
        }
    }

    printf("Distinct elements: ");
    for (int i = 0; i < count; i++) {
        printf("%d ", found[i]);
    }
    printf("\n");
}
```

```
int main() {
    int arr[] = {12, 10, 9, 45, 2, 10, 10, 45};
    int n = sizeof(arr) / sizeof(arr[0]);
    printDistinct(arr, n);
    return 0;
}
```

- Write a program to count the total number of nonzero elements in a two- dimensional array.

```
#include <stdio.h>

int main() {
    int rows, cols;
    printf("Enter number of rows and columns: ");
    scanf("%d %d", &rows, &cols);
    int matrix[rows][cols];
    int count = 0;

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

    printf("Total nonzero elements: %d\n", count);
    return 0;
}
```

- Write a program using pointers to interchange the second biggest and the second smallest number in the array.\

```
#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

void interchangeSecond(int arr[], int n) {
    int firstMin = 0, secondMin = 0, firstMax = 0, secondMax = 0;

    for (int i = 0; i < n; i++) {
        if (arr[i] < arr[firstMin]) {
            secondMin = firstMin;
            firstMin = i;
        } else if (arr[i] < arr[secondMin] && arr[i] != arr[firstMin]) {
            secondMin = i;
        }

        if (arr[i] > arr[firstMax]) {
            secondMax = firstMax;
            firstMax = i;
        } else if (arr[i] > arr[secondMax] && arr[i] != arr[firstMax]) {
            secondMax = i;
```

```
        }
    }

    if (secondMin != firstMin && secondMax != firstMax) {
        swap(&arr[secondMin], &arr[secondMax]);
    }

    printf("Array after swapping: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    int arr[] = {3, 5, 1, 2, 4, 6};
    int n = sizeof(arr) / sizeof(arr[0]);
    interchangeSecond(arr, n);
    return 0;
}
```