1. Matrix Addition (Using Basic Arrays)

```
#include <stdio.h>

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

    int matrix1[m][n], matrix2[m][n], result[m][n];

    // Input first matrix
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &matrix1[i][j]);
        }
    }

    // Input second matrix
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &matrix2[i][j]);
        }
    }

    // Adding matrices
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }

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

    return 0;
}
```

2. Removing Duplicates and Sorting

```c
#include <stdio.h>

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

    int arr[n], unique[n], size = 0;

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

    // Remove duplicates
    for (int i = 0; i < n; i++) {
        int flag = 0;
        for (int j = 0; j < size; j++) {
            if (arr[i] == unique[j]) {
                flag = 1;
                break;
            }
        }
        if (!flag) {
            unique[size++] = arr[i];
        }
    }

    // Sort the unique array
    for (int i = 0; i < size-1; i++) {
        for (int j = i+1; j < size; j++) {
            if (unique[i] > unique[j]) {
                int temp = unique[i];
                unique[i] = unique[j];
                unique[j] = temp;
            }
        }
    }

    // Output sorted unique numbers
    for (int i = 0; i < size; i++) {
        printf("%d ", unique[i]);
    }
    printf("\n");

    return 0;
}
```

TOTAL NO. Of Digits

```
#include <stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int count=0;
    while(n>0){
      int digits=n%10;
      count++;
      n/=10;
    }
    printf("%d",count);
}
```

3. Factorial Calculator

```
#include <stdio.h>

long long factorial(int n) {
    long long fact = 1;
    for (int i = 1; i <= n; i++) {
        fact *= i;
    }
    return fact;
}

int trailingZeroes(int n) {
    int count = 0;
    while (n >= 5) {
        count += n / 5;
        n /= 5;
    }
    return count;
}

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

    long long fact = factorial(n);

    printf("%lld\n", fact);
    printf("%d\n", trailingZeroes(n));
    printf(fact % n == 0 ? "Y\n" : "N\n");

    // Count digits
    int digits = 0;
    long long temp = fact;
    while (temp > 0) {
        digits++;
        temp /= 10;
    }
    printf("%d\n", digits);

    return 0;
}
```

4. **Shopping Cart System**

```
#include <stdio.h>

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

    int total = 0, price;
    char name[100];

    // Input items
    for (int i = 0; i < n; i++) {
        scanf("%s %d", name, &price);
        total += price;
    }

    // Check total
    if (total > 100000) {
        printf("Error: Total exceeds 100000 limit\n");
        printf("Please remove some items\n");
    } else {
        printf("Total: %d\n", total);
    }

    return 0;
}
```

5. **Insurance Premium Calculation**

```
#include <stdio.h>

int main() {
    int age, smokingStatus, preExistingConditions;
    scanf("%d %d %d", &age, &smokingStatus, &preExistingConditions);

    int premium = 500;

    if (age > 50) {
        premium += 200;
    } else if (age >= 31) {
        premium += 100;
    }

    if (smokingStatus == 1) {
        premium += 150;
    }

    if (preExistingConditions == 1) {
        premium += 300;
    }

    printf("%d\n", premium);

    return 0;
}
```

6. Cashback Calculation

```
#include <stdio.h>

int main() {
    int orderID;
    double amount, discount;
    char paymentMethod;

    scanf("%d %lf %lf %c", &orderID, &amount, &discount, &paymentMethod);

    double finalAmount = amount - discount;
    double cashback = 0;

    if (paymentMethod == 'C') {
        cashback = finalAmount * 0.10;
    } else if (paymentMethod == 'D') {
        cashback = finalAmount * 0.05;
    }

    printf("ID: %d, Final Amount: %.2f, Cashback: %.2f\n", orderID,
finalAmount, cashback);

    return 0;
}
```

7. Count Digits Greater Than a Threshold

```
#include <stdio.h>

int greater(int N, int X) {
    int count = 0;
    while (N > 0) {
        int digit = N % 10;
        if (digit > X) {
            count++;
        }
        N /= 10;
    }
    return count;
}

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

    int result = greater(N, X);
    printf("Number of digits greater than %d: %d\n", X, result);

    return 0;
}
```

8. Matrix Operations (Adding Two Matrices)

```
#include <stdio.h>

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

    int matrix1[m][n], matrix2[m][n], result[m][n];

    // Input first matrix
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &matrix1[i][j]);
        }
    }

    // Input second matrix
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &matrix2[i][j]);
        }
    }

    // Adding matrices
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }

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

    return 0;
}
```