31. wap_to_check_whether_the_entered_number_is_armstrong_or_not

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

int main() {
   int num, original, remainder, count = 0;
   printf("Enter a number: ");
   scanf("%d", &num);
   original = num;

   while (original != 0) {
      original /= 10;
      count++;
   }

   original = num;
   int sum = 0;
   while (original != 0) {
      remainder = original % 10;
      sum += pow(remainder, count);
      original /= 10;
   }

   if (sum == num) {
      printf("Armstrong\n");
   } else {
      printf("Not an Armstrong\n");
```

```
    }
    return 0;
}
```

32. wap_to_accept_a_2_d_array_and_display_it

```
#include <stdio.h>

int main() {
    int arr[10][10], rows, cols;
    printf("Enter number of rows and columns: ");
    scanf("%d %d", &rows, &cols);
    printf("Enter elements:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &arr[i][j]);
        }
    }
    printf("Array elements:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

33. wap_to_display_sum_of_the_diagonal_elements_of_the_matrix

```c
#include <stdio.h>

int main() {
   int arr[10][10], n, sum = 0;
   printf("Enter the size of the square matrix: ");
   scanf("%d", &n);
   printf("Enter elements:\n");
   for (int i = 0; i < n; i++) {
      for (int j = 0; j < n; j++) {
         scanf("%d", &arr[i][j]);
         if (i == j) {
            sum += arr[i][j];
         }
      }
   }
   printf("Sum of diagonal elements: %d\n", sum);
   return 0;
}
```

34. wap_to_display_transpose_of_the_matrix

```c
#include <stdio.h>

int main() {
   int arr[10][10], trans[10][10], rows, cols;
   printf("Enter number of rows and columns: ");
   scanf("%d %d", &rows, &cols);
   printf("Enter elements:\n");
```

```
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &arr[i][j]);
            trans[j][i] = arr[i][j];
        }
    }
    printf("Transpose matrix:\n");
    for (int i = 0; i < cols; i++) {
        for (int j = 0; j < rows; j++) {
            printf("%d ", trans[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

35. wap_to_display_upper_triangular_matrix

```
#include <stdio.h>

int main() {
    int arr[10][10], rows, cols;
    printf("Enter number of rows and columns: ");
    scanf("%d %d", &rows, &cols);
    printf("Enter elements:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &arr[i][j]);
        }
```

```
    }
    printf("Upper triangular matrix:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (j < i) {
                printf("0 ");
            } else {
                printf("%d ", arr[i][j]);
            }
        }
        printf("\n");
    }
    return 0;
}
```

36. wap_to_display_matrix_addition

```
#include <stdio.h>

int main() {
    int arr1[10][10], arr2[10][10], sum[10][10], rows, cols;
    printf("Enter number of rows and columns: ");
    scanf("%d %d", &rows, &cols);
    printf("Enter elements of first matrix:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &arr1[i][j]);
        }
    }
```

```
    printf("Enter elements of second matrix:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &arr2[i][j]);
        }
    }
    printf("Sum of matrices:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            sum[i][j] = arr1[i][j] + arr2[i][j];
            printf("%d ", sum[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

37. wap_to_display_marksheet_of_10_students

```
#include <stdio.h>

int main() {
    int marks[10];
    printf("Enter marks of 10 students:\n");
    for (int i = 0; i < 10; i++) {
        printf("Student %d: ", i + 1);
        scanf("%d", &marks[i]);
    }
    printf("Marksheet:\n");
```

```
    for (int i = 0; i < 10; i++) {
        printf("Student %d: %d\n", i + 1, marks[i]);
    }
    return 0;
}
```

38. wap_menu_driven_to_create_user_defined_functions

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

int factorial(int n) {
    return (n == 0 || n == 1) ? 1 : n * factorial(n - 1);
}

int is_prime(int n) {
    if (n <= 1) return 0;
    for (int i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) return 0;
    }
    return 1;
}

int sum_of_digits(int n) {
    int sum = 0;
    while (n != 0) {
        sum += n % 10;
        n /= 10;
    }
```

```
    return sum;
}

int is_palindrome(int n) {
    int original = n, reversed = 0;
    while (n != 0) {
        reversed = reversed * 10 + n % 10;
        n /= 10;
    }
    return original == reversed;
}

int reverse_number(int n) {
    int reversed = 0;
    while (n != 0) {
        reversed = reversed * 10 + n % 10;
        n /= 10;
    }
    return reversed;
}

int is_armstrong(int n) {
    int original = n, sum = 0, count = 0;
    while (original != 0) {
        original /= 10;
        count++;
    }
    original = n;
    while (original != 0) {
```

```
        sum += pow(original % 10, count);
        original /= 10;
    }
    return sum == n;
}

int main() {
    int choice, num;
    printf("Menu:\n1. Factorial\n2. Prime\n3. Sum of digits\n4. Palindrome\n5. Reverse\n6. Armstrong\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);
    printf("Enter a number: ");
    scanf("%d", &num);

    switch (choice) {
        case 1: printf("Factorial: %d\n", factorial(num)); break;
        case 2: printf(is_prime(num) ? "Prime\n" : "Not Prime\n"); break;
        case 3: printf("Sum of digits: %d\n", sum_of_digits(num)); break;
        case 4: printf(is_palindrome(num) ? "Palindrome\n" : "Not Palindrome\n"); break;
        case 5: printf("Reversed number: %d\n", reverse_number(num)); break;
        case 6: printf(is_armstrong(num) ? "Armstrong\n" : "Not Armstrong\n"); break;
        default: printf("Invalid choice\n");
    }
    return 0;
}
```