

Ejercicio 1: Promedio de un arreglo de números enteros positivos

```
void inicializarArreglo(int arreglo[], int tam) {
    for (int i = 0; i < tam; i++) {
        arreglo[i] = -1;
    }
}

void ingresarValores(int arreglo[], int tam) {
    int valor;
    int i = 0;

    printf("Ingrese valores positivos (-1 para terminar):\n");
    do {
        scanf("%d", &valor);
        if (valor >= 0) {
            arreglo[i] = valor;
            i++;
        }
    } while (valor >= 0 && i < tam);
}

int sumarContenido(int arreglo[], int tam) {
    int suma = 0;
    for (int i = 0; i < tam; i++) {
        suma += arreglo[i];
    }
    return suma;
}

int contarValores(int arreglo[], int tam) {
    int contador = 0;
    for (int i = 0; i < tam; i++) {
        if (arreglo[i] != -1) {
            contador++;
        }
    }
    return contador;
}

float calcularPromedio(int arreglo[], int tam) {
    int suma = sumarContenido(arreglo, tam);
    int cantidad = contarValores(arreglo, tam);
    return (float)suma / cantidad;
}

int main() {
    const int tam = 100; // Tamaño del arreglo
    int valores[tam];

    inicializarArreglo(valores, tam);
    ingresarValores(valores, tam);

    float promedio = calcularPromedio(valores, tam);

    printf("El promedio de los valores ingresados es: %.2f\n", promedio);

    return 0;
}
```

Ejercicio 2: Invertir el contenido de un arreglo de caracteres

```
void cargarArreglo(char arreglo[], int tam) {
    printf("Ingrese una cadena de hasta %d caracteres: ", tam);
    scanf("%s", arreglo);
}

void invertirArreglo(char arreglo[], int tam) {
    int i = 0;
    int j = strlen(arreglo) - 1;

    while (i < j) {
        char temp = arreglo[i];
```

```

        arreglo[i] = arreglo[j];
        arreglo[j] = temp;

        i++;
        j--;
    }
}

void mostrarArreglo(char arreglo[], int tam) {
    printf("Arreglo invertido: %s\n", arreglo);
}

int main() {
    const int tam = 100; // Tamaño del arreglo
    char cadena[tam];

    cargarArreglo(cadena, tam);
    invertirArreglo(cadena, tam);
    mostrarArreglo(cadena, tam);

    return 0;
}

```

Ejercicio 3: Convertir arreglo de dígitos en valor decimal

```

int arregloADecimal(int arreglo[], int tam) {
    int decimal = 0;
    for (int i = 0; i < tam; i++) {
        decimal = decimal * 10 + arreglo[i];
    }
    return decimal;
}

int main() {
    const int tam = 4; // Tamaño del arreglo
    int digitos[tam];

    printf("Ingrese %d dígitos:\n", tam);
    for (int i = 0; i < tam; i++) {
        scanf("%d", &digitos[i]);
    }
    int valorDecimal = arregloADecimal(digitos, tam);
    printf("Valor decimal resultante: %d\n", valorDecimal);
    return 0;
}

```

Ejercicio 4: Posición del valor máximo en un arreglo de números enteros

```

int posicionMaximo(int arreglo[], int tam) {
    int maximo = arreglo[0];
    int posMaximo = 0;

    for (int i = 1; i < tam; i++) {
        if (arreglo[i] > maximo) {
            maximo = arreglo[i];
            posMaximo = i;
        }
    }

    return posMaximo;
}

int main() {
    const int tam = 10; // Tamaño del arreglo
    int numeros[tam];

    printf("Ingrese %d números enteros:\n", tam);
    for (int i = 0; i < tam; i++) {
        scanf("%d", &numeros[i]);
    }

    int posicion = posicionMaximo(numeros, tam);
}

```

```

printf("La posición del valor máximo es: %d\n", posicion);

return 0;
}

```

Ejercicio 5: Máxima precipitación por mes en una matriz

```

void cargarMatriz(int matriz[][31], int filas, int columnas) {
    // Código para cargar la matriz de forma automática (puedes modificarlo)
    for (int i = 0; i < filas; i++) {
        for (int j = 0; j < columnas; j++) {
            matriz[i][j] = rand() % 100; // Valores aleatorios entre 0 y 99
        }
    }
}

int diaMaximaPrecipitacion(int matriz[][31], int mes, int columnas) {
    int maximo = matriz[mes - 1][0];
    int diaMaximo = 1;

    for (int i = 1; i < columnas; i++) {
        if (matriz[mes - 1][i] > maximo) {
            maximo = matriz[mes - 1][i];
            diaMaximo = i + 1;
        }
    }

    return diaMaximo;
}

void cargarMaximosPorMes(int matriz[][31], int filas, int columnas, int maximos[]) {
    for (int i = 0; i < filas; i++) {
        maximos[i] = diaMaximaPrecipitacion(matriz, i + 1, columnas);
    }
}

int main() {
    const int filas = 12;
    const int columnas = 31;
    int matriz[filas][columnas];
    int maximosPorMes[filas];

    cargarMatriz(matriz, filas, columnas);
    cargarMaximosPorMes(matriz, filas, columnas, maximosPorMes);

    printf("Días de máxima precipitación por mes:\n");
    for (int i = 0; i < filas; i++) {
        printf("Mes %d: Día %d\n", i + 1, maximosPorMes[i]);
    }

    return 0;
}

```

Ejercicio 6: Verificar si una cadena es palíndromo (capicua)

```

int esPalindromo(char cadena[]) {
    int i = 0;
    int j = strlen(cadena) - 1;

    while (i < j) {
        if (cadena[i] != cadena[j]) {
            return 0; // No es palíndromo
        }
        i++;
        j--;
    }

    return 1; // Es palíndromo
}

```

```

int main() {
    const int tam = 100; // Tamaño de la cadena
    char cadena[tam];

    printf("Ingrese una cadena: ");
    scanf("%s", cadena);

    if (esPalindromo(cadena)) {
        printf("La cadena es un palíndromo.\n");
    } else {
        printf("La cadena no es un palíndromo.\n");
    }

    return 0;
}

```

Ejercicio 7: Intercambiar el contenido de dos variables (por referencia)

```

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

int main() {
    int num1, num2;

    printf("Ingrese dos números enteros:\n");
    scanf("%d %d", &num1, &num2);

    intercambiar(&num1, &num2);

    printf("Números intercambiados: %d y %d\n", num1, num2);

    return 0;
}

```

Ejercicio 8: Interclasificación de dos arreglos ordenados (intercalarlos)

```

void interclasificar(int arr1[], int tam1, int arr2[], int tam2, int resultado[]) {
    int i = 0, j = 0, k = 0;

    while (i < tam1 && j < tam2) {
        if (arr1[i] < arr2[j]) {
            resultado[k] = arr1[i];
            i++;
        } else {
            resultado[k] = arr2[j];
            j++;
        }
        k++;
    }

    while (i < tam1) {
        resultado[k] = arr1[i];
        i++;
        k++;
    }

    while (j < tam2) {
        resultado[k] = arr2[j];
        j++;
        k++;
    }
}

void mostrarArreglo(int arreglo[], int tam) {

```

```

    for (int i = 0; i < tam; i++) {
        printf("%d ", arreglo[i]);
    }
    printf("\n");
}

int main() {
    const int tam1 = 5;
    const int tam2 = 7;
    int arr1[tam1] = {1, 3, 5, 7, 9};
    int arr2[tam2] = {2, 4, 6, 8, 10, 12, 14};
    int resultado[tam1 + tam2];

    interclasificar(arr1, tam1, arr2, tam2, resultado);
    mostrarArreglo(resultado, tam1 + tam2);

    return 0;
}

```

Ejercicio 9: Encontrar el apellido de la persona de mayor edad

```

void encontrarMayorEdad(char apellidos[][100], int edades[], int tam, char apellidoMayor[]) {
    int maxEdad = edades[0];
    int posMaxEdad = 0;

    for (int i = 1; i < tam; i++) {
        if (edades[i] > maxEdad) {
            maxEdad = edades[i];
            posMaxEdad = i;
        }
    }

    strcpy(apellidoMayor, apellidos[posMaxEdad]);
}

int main() {
    const int tam = 5;
    char apellidos[tam][100];
    int edades[tam];
    char apellidoMayor[100];

    printf("Ingrese los apellidos y edades:\n");
    for (int i = 0; i < tam; i++) {
        scanf("%s %d", apellidos[i], &edades[i]);
    }

    encontrarMayorEdad(apellidos, edades, tam, apellidoMayor);

    printf("El apellido de la persona de mayor edad es: %s\n", apellidoMayor);

    return 0;
}

```

Ejercicio 10: Función para calcular máximo, mínimo y promedio

```

void calcularMaxMinPromedio(int arreglo[], int tam, int *maximo, int *minimo, float *promedio) {
    *maximo = arreglo[0];
    *minimo = arreglo[0];
    int suma = arreglo[0];

    for (int i = 1; i < tam; i++) {
        if (arreglo[i] > *maximo) {
            *maximo = arreglo[i];
        }
        if (arreglo[i] < *minimo) {
            *minimo = arreglo[i];
        }
        suma += arreglo[i];
    }
}

```

```
    *promedio = (float)suma / tam;
}

int main() {
    const int tam = 5;
    int numeros[tam];

    printf("Ingrese %d números enteros:\n", tam);
    for (int i = 0; i < tam; i++) {
        scanf("%d", &numeros[i]);
    }

    int maximo, minimo;
    float promedio;

    calcularMaxMinPromedio(numeros, tam, &maximo, &minimo, &promedio);

    printf("Máximo: %d\n", maximo);
    printf("Mínimo: %d\n", minimo);
    printf("Promedio: %.2f\n", promedio);

    return 0;
}
```