Punto 4

```
#include <stdio.h>

int main() {
    float ancho, largo, area;
    char unidad[10];

    // Solicitar al usuario ingresar el ancho de la habitación
    printf("Ingrese el ancho de la habitación: ");
    scanf("%f", &ancho);

    // Solicitar al usuario ingresar la unidad (pies o metros)
    printf("Ingrese la unidad (pies o metros): ");
    scanf("%s", unidad);

    // Solicitar al usuario ingresar el largo de la habitación
    printf("Ingrese el largo de la habitación: ");
    scanf("%f", &largo);

    // Calcular el área de la habitación
    area = ancho * largo;

    // Mostrar el área con la unidad especificada por el usuario
    printf("El área de la habitación es %.2f %s cuadrados.\n", area, unidad);

    return 0;
}
```

Punto 5

```
#include <stdio.h>

int main() {
    float longitud, ancho, area_acres;

    // Solicitar al usuario ingresar la longitud del campo en pies
    printf("Ingrese la longitud del campo en pies: ");
    scanf("%f", &longitud);

    // Solicitar al usuario ingresar el ancho del campo en pies
    printf("Ingrese el ancho del campo en pies: ");
    scanf("%f", &ancho);

    // Calcular el área del campo en acres
    float area_pies_cuadrados = longitud * ancho;
    area_acres = area_pies_cuadrados / 43560.0; // 1 acre = 43,560 pies cuadrados

    // Mostrar el área del campo en acres
    printf("El área del campo es %.2f acres.\n", area_acres);

    return 0;
}
```

Punto 6

```
int main() {
    // Definir los vectores y matrices
    int u[3] = {1, 2, 3};
    int v[3] = {6, 5, 4};
    int A[3][3] = {{1, 5, 0}, {7, 1, 2}, {0, 0, 1}};
    int B[3][3] = {{-2, 0, 1}, {1, 0, 0}, {4, 1, 0}};

    // 1. w = u - 3v
    int w[3];
    for (int i = 0; i < 3; i++) {
        w[i] = u[i] - 3 * v[i];
    }

    // 2. x = u - v
    int x[3];
    for (int i = 0; i < 3; i++) {
        x[i] = u[i] - v[i];
    }

    // 3. y = Au
    int y[3];
    for (int i = 0; i < 3; i++) {
        y[i] = 0;
        for (int j = 0; j < 3; j++) {
            y[i] += A[i][j] * u[j];
        }
    }

    // 4. z = Au - v
```

```
int z[3];
for (int i = 0; i < 3; i++) {
    z[i] = y[i] - v[i];
}

// 5. C = 4A - 3B
int C[3][3];
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        C[i][j] = 4 * A[i][j] - 3 * B[i][j];
    }
}

// 6. D = AB
int D[3][3];
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        D[i][j] = 0;
        for (int k = 0; k < 3; k++) {
            D[i][j] += A[i][k] * B[k][j];
        }
    }
}

// Mostrar los resultados
printf("1. w = %d %d %d\n", w[0], w[1], w[2]);
printf("2. x = %d %d %d\n", x[0], x[1], x[2]);
printf("3. y = %d %d %d\n", y[0], y[1], y[2]);
printf("4. z = %d %d %d\n", z[0], z[1], z[2]);
printf("5. C =\n");
```

```
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ", C[i][j]);
        }
        printf("\n");
    }
    printf("6. D =\n");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ", D[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Punto 7

```
#include <stdio.h>

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

int main() {
    int num1 = 5;
    int num2 = 10;
```

```
    printf("Valores iniciales:\n");
    printf("num1 = %d\n", num1);
    printf("num2 = %d\n", num2);

    // Llamar a la función para intercambiar los valores
    intercambiar(&num1, &num2);

    printf("Valores intercambiados:\n");
    printf("num1 = %d\n", num1);
    printf("num2 = %d\n", num2);

    return 0;
}
```

Punto 8

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

int main() {
    int dimension;

    // Solicitar la dimensión de los vectores al usuario
    printf("Ingrese la dimensión de los vectores: ");
    scanf("%d", &dimension);

    // Asignar memoria dinámica para los vectores
    double *vector1 = (double *)malloc(dimension * sizeof(double));
    double *vector2 = (double *)malloc(dimension * sizeof(double));

    // Comprobar si se asignó memoria correctamente
```

```
if (vector1 == NULL || vector2 == NULL) {
    printf("Error al asignar memoria para los vectores.\n");
    return 1;
}

// Pedir al usuario los valores de los elementos de ambos vectores
printf("Ingrese los valores del primer vector:\n");
for (int i = 0; i < dimension; i++) {
    printf("Elemento %d: ", i + 1);
    scanf("%lf", &vector1[i]);
}

printf("Ingrese los valores del segundo vector:\n");
for (int i = 0; i < dimension; i++) {
    printf("Elemento %d: ", i + 1);
    scanf("%lf", &vector2[i]);
}

// Calcular el producto escalar
double producto_escalar = 0.0;
for (int i = 0; i < dimension; i++) {
    producto_escalar += vector1[i] * vector2[i];
}

// Imprimir el resultado del producto escalar
printf("El producto escalar de los vectores es: %lf\n", producto_escalar);

// Liberar la memoria asignada para los vectores
free(vector1);
free(vector2);
```

```
    return 0;
}
```

Punto 9

```
#include <stdio.h>

int funcion(double *suma, double *producto, double x, double y) {
    if (suma == NULL || producto == NULL) {
        // Verificar si los punteros son nulos
        return 0; // Error: punteros nulos
    }

    *suma = x + y;
    *producto = x * y;

    return 1; // Éxito
}

int main() {
    double x = 5.0;
    double y = 3.0;
    double resultado_suma, resultado_producto;

    if (funcion(&resultado_suma, &resultado_producto, x, y)) {
        printf("La suma es: %.2lf\n", resultado_suma);
        printf("El producto es: %.2lf\n", resultado_producto);
    } else {
        printf("Error en la función.\n");
    }
```

```
    return 0;
}
```

Punto 10

```
#include <stdio.h>

int main() {
    double matriz[2][2];
    double determinante;

    // Solicitar al usuario los elementos de la matriz
    printf("Ingrese los elementos de la matriz 2x2:\n");
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            printf("Elemento (%d, %d): ", i + 1, j + 1);
            scanf("%lf", &matriz[i][j]);
        }
    }

    // Calcular el determinante de la matriz
    determinante = matriz[0][0] * matriz[1][1] - matriz[0][1] * matriz[1][0];

    // Verificar si el determinante es igual a cero (no tiene inversa)
    if (determinante == 0) {
        printf("La matriz no tiene inversa porque su determinante es cero.\n");
    } else {
        // Calcular la matriz inversa
        double inversa[2][2];
        inversa[0][0] = matriz[1][1] / determinante;
```

```
        inversa[0][1] = -matriz[0][1] / determinante;
        inversa[1][0] = -matriz[1][0] / determinante;
        inversa[1][1] = matriz[0][0] / determinante;

        // Imprimir la matriz inversa
        printf("La matriz inversa es:\n");
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                printf("%.2lf\t", inversa[i][j]);
            }
            printf("\n");
        }
    }

    return 0;
}
```