1)

#include <stdio.h>

#define N 6

int main() {

// Declaración de variables

int vector[N];

// Ingreso de datos

printf("Ingrese 6 elementos para el vector:\n");

for (int i = 0; i < N; i++) {

printf("Elemento %d: ", i + 1);

scanf("%d", &vector[i]);

}

// Mostrar contenido del vector

printf("\nContenido del vector:\n");

for (int i = 0; i < N; i++) {

printf("%d ", vector[i]);

}

printf("\n");

return 0;

}

2)

#include <stdio.h>

#define N 7

int main() {

int vector[N];

int mayor, posicion;

printf("Ingrese 7 números enteros positivos:\n");

for (int i = 0; i < N; i++) {

printf("Elemento %d: ", i + 1);

scanf("%d", &vector[i]);

if (i == 0 || vector[i] > mayor) {

mayor = vector[i];

posicion = i + 1;

}

}

printf("\nContenido del vector:\n");

for (int i = 0; i < N; i++) {

printf("%d ", vector[i]);

}

printf("\n\nEl mayor elemento es %d y se encuentra en la posición %d.\n", mayor, posicion);

return 0;

}

3)

#include <stdio.h>

#define MAX\_COMPRAS 10

int main() {

float gastos[MAX\_COMPRAS];

float total = 0, mayorGasto = 0;

int numCompras;

do {

printf("Ingrese el número de compras (máximo 10): ");

scanf("%d", &numCompras);

} while (numCompras <= 0 || numCompras > MAX\_COMPRAS);

printf("Ingrese los gastos en compras:\n");

for (int i = 0; i < numCompras; i++) {

printf("Compra %d: $", i + 1);

scanf("%f", &gastos[i]);

total += gastos[i];

if (i == 0 || gastos[i] > mayorGasto) {

mayorGasto = gastos[i];

}

}

printf("\nMonto total gastado: $%.2f\n", total);

printf("El mayor gasto fue: $%.2f\n", mayorGasto);

return 0;

}

4)

#include <stdio.h>

#define N 8

int main() {

int numeros[N];

int num1, num2;

printf("Ingrese 8 números enteros:\n");

for (int i = 0; i < N; i++) {

printf("Número %d: ", i + 1);

scanf("%d", &numeros[i]);

}

printf("\nIngrese dos números adicionales:\n");

printf("Número 1: ");

scanf("%d", &num1);

printf("Número 2: ");

scanf("%d", &num2);

int conteo1 = 0, conteo2 = 0;

int posicion1 = -1, posicion2 = -1;

for (int i = 0; i < N; i++) {

if (numeros[i] == num1) {

conteo1++;

posicion1 = i + 1;

}

if (numeros[i] == num2) {

conteo2++;

posicion2 = i + 1;

}

}

if (conteo1 > 0) {

printf("\n%d pertenece a la secuencia y aparece %d veces en la posición %d.\n", num1, conteo1, posicion1);

} else {

printf("\n%d no pertenece a la secuencia.\n", num1);

}

if (conteo2 > 0) {

printf("%d pertenece a la secuencia y aparece %d veces en la posición %d.\n", num2, conteo2, posicion2);

} else {

printf("%d no pertenece a la secuencia.\n", num2);

}

return 0;

}

5)

#include <stdio.h>

#define MAX\_NUMEROS 20

int main() {

int numeros[MAX\_NUMEROS];

int pares[MAX\_NUMEROS], impares[MAX\_NUMEROS];

int cantidadNumeros = 0, cantidadPares = 0, cantidadImpares = 0;

int numero;

printf("Ingrese números enteros (0 para finalizar o máximo 20 números):\n");

do {

printf("Número %d: ", cantidadNumeros + 1);

scanf("%d", &numero);

if (numero != 0) {

numeros[cantidadNumeros] = numero;

if (numero % 2 == 0) {

pares[cantidadPares] = numero;

cantidadPares++;

} else {

impares[cantidadImpares] = numero;

cantidadImpares++;

}

cantidadNumeros++;

}

} while (numero != 0 && cantidadNumeros < MAX\_NUMEROS);

printf("\nNúmeros pares ingresados:\n");

for (int i = 0; i < cantidadPares; i++) {

printf("%d ", pares[i]);

}

printf("\n\nNúmeros impares ingresados:\n");

for (int i = 0; i < cantidadImpares; i++) {

printf("%d ", impares[i]);

}

return 0;

}

6)

#include <stdio.h>

#define TAMANO 10

int main() {

int arreglo[TAMANO];

int maximo, minimo;

int ocurrenciasMaximo = 0, ocurrenciasMinimo = 0;

printf("Ingrese 10 números enteros para el arreglo:\n");

for (int i = 0; i < TAMANO; i++) {

printf("Elemento %d: ", i + 1);

scanf("%d", &arreglo[i]);

if (i == 0) {

maximo = arreglo[i];

minimo = arreglo[i];

} else {

if (arreglo[i] > maximo) {

maximo = arreglo[i];

} else if (arreglo[i] < minimo) {

minimo = arreglo[i];

}

}

}

for (int i = 0; i < TAMANO; i++) {

if (arreglo[i] == maximo) {

ocurrenciasMaximo++;

} else if (arreglo[i] == minimo) {

ocurrenciasMinimo++;

}

}

printf("\nValor máximo: %d\n", maximo);

printf("Número de ocurrencias del valor máximo: %d\n", ocurrenciasMaximo);

printf("Valor mínimo: %d\n", minimo);

printf("Número de ocurrencias del valor mínimo: %d\n", ocurrenciasMinimo);

return 0;

}

7)

#include <stdio.h>

#define NUM\_CORREDORES 10

int main() {

float tiempos[NUM\_CORREDORES];

float tiempoMedio = 0;

int primerPuesto, segundoPuesto, ultimoPuesto;

printf("Ingrese los tiempos de los corredores:\n");

for (int i = 0; i < NUM\_CORREDORES; i++) {

printf("Tiempo del corredor %d: ", i + 1);

scanf("%f", &tiempos[i]);

tiempoMedio += tiempos[i];

}

tiempoMedio /= NUM\_CORREDORES;

primerPuesto = segundoPuesto = ultimoPuesto = 0;

for (int i = 1; i < NUM\_CORREDORES; i++) {

if (tiempos[i] < tiempos[primerPuesto]) {

segundoPuesto = primerPuesto;

primerPuesto = i;

} else if (tiempos[i] < tiempos[segundoPuesto]) {

segundoPuesto = i;

}

if (tiempos[i] > tiempos[ultimoPuesto]) {

ultimoPuesto = i;

}

}

printf("\nCorredor del primer puesto: Corredor %d con tiempo %.2f\n", primerPuesto + 1, tiempos[primerPuesto]);

printf("Corredor del segundo puesto: Corredor %d con tiempo %.2f\n", segundoPuesto + 1, tiempos[segundoPuesto]);

printf("Corredor del último puesto: Corredor %d con tiempo %.2f\n", ultimoPuesto + 1, tiempos[ultimoPuesto]);

printf("Tiempo medio de la carrera: %.2f\n", tiempoMedio);

return 0;

}

8)

#include <stdio.h>

#define N 7

void intercambiar(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

void ordenarVector(int vector[], int tamano) {

for (int i = 0; i < tamano - 1; i++) {

for (int j = 0; j < tamano - i - 1; j++) {

if (vector[j] > vector[j + 1]) {

intercambiar(&vector[j], &vector[j + 1]);

}

}

}

}

int main() {

int vector[N];

printf("Ingrese 7 números enteros positivos:\n");

for (int i = 0; i < N; i++) {

printf("Elemento %d: ", i + 1);

scanf("%d", &vector[i]);

}

ordenarVector(vector, N);

printf("\nContenido del vector ordenado de menor a mayor:\n");

for (int i = 0; i < N; i++) {

printf("%d ", vector[i]);

}

printf("\n");

return 0;

}

9)

#include <stdio.h>

#define N 8

void intercambiar(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

void ordenarVectorDescendente(int vector[], int tamano) {

for (int i = 0; i < tamano - 1; i++) {

for (int j = 0; j < tamano - i - 1; j++) {

if (vector[j] < vector[j + 1]) {

intercambiar(&vector[j], &vector[j + 1]);

}

}

}

}

int main() {

int vector[N];

printf("Ingrese 8 números enteros positivos:\n");

for (int i = 0; i < N; i++) {

printf("Elemento %d: ", i + 1);

scanf("%d", &vector[i]);

}

printf("\nVector ingresado:\n");

for (int i = 0; i < N; i++) {

printf("%d ", vector[i]);

}

ordenarVectorDescendente(vector, N);

printf("\nVector ordenado en forma decreciente:\n");

for (int i = 0; i < N; i++) {

printf("%d ", vector[i]);

}

printf("\n");

return 0;

}

10)

#include <stdio.h>

#define N 8

void intercambiar(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

void ordenarPositivosDescendente(int vector[], int tamano) {

for (int i = 0; i < tamano - 1; i++) {

for (int j = 0; j < tamano - i - 1; j++) {

if (vector[j] < vector[j + 1]) {

intercambiar(&vector[j], &vector[j + 1]);

}

}

}

}

void ordenarNegativosCreciente(int vector[], int tamano) {

for (int i = 0; i < tamano - 1; i++) {

for (int j = 0; j < tamano - i - 1; j++) {

if (vector[j] > vector[j + 1]) {

intercambiar(&vector[j], &vector[j + 1]);

}

}

}

}

int main() {

int vector[N];

printf("Ingrese 8 números enteros (distintos de cero, positivos y negativos):\n");

for (int i = 0; i < N; i++) {

do {

printf("Elemento %d: ", i + 1);

scanf("%d", &vector[i]);

} while (vector[i] == 0);

}

printf("\nVector ingresado:\n");

for (int i = 0; i < N; i++) {

printf("%d ", vector[i]);

}

printf("\nPositivos ordenados en forma decreciente:\n");

int positivos[N], cantidadPositivos = 0;

for (int i = 0; i < N; i++) {

if (vector[i] > 0) {

positivos[cantidadPositivos] = vector[i];

cantidadPositivos++;

}

}

ordenarPositivosDescendente(positivos, cantidadPositivos);

for (int i = 0; i < cantidadPositivos; i++) {

printf("%d ", positivos[i]);

}

printf("\nNegativos ordenados en forma creciente:\n");

int negativos[N], cantidadNegativos = 0;

for (int i = 0; i < N; i++) {

if (vector[i] < 0) {

negativos[cantidadNegativos] = vector[i];

cantidadNegativos++;

}

}

ordenarNegativosCreciente(negativos, cantidadNegativos);

for (int i = 0; i < cantidadNegativos; i++) {

printf("%d ", negativos[i]);

}

printf("\n");

return 0;

}

11)

#include <stdio.h>

#include <string.h>

#define MAX\_LONGITUD 100

int main() {

char cadena[MAX\_LONGITUD];

printf("Ingrese una cadena de caracteres: ");

fgets(cadena, sizeof(cadena), stdin);

size\_t longitud = strlen(cadena);

if (longitud > 0 && cadena[longitud - 1] == '\n') {

cadena[longitud - 1] = '\0';

}

for (int i = 0, j = longitud - 1; i < j; i++, j--) {

char temp = cadena[i];

cadena[i] = cadena[j];

cadena[j] = temp;

}

printf("Cadena invertida: %s\n", cadena);

return 0;

}

12)

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#define MAX\_LONGITUD 100

void eliminarEspacios(char cadena[]) {

int i, j;

for (i = 0, j = 0; i < strlen(cadena); i++) {

if (!isspace(cadena[i])) {

cadena[j++] = cadena[i];

}

}

cadena[j] = '\0';

}

int esPalindromo(char cadena[]) {

int longitud = strlen(cadena);

for (int i = 0, j = longitud - 1; i < j; i++, j--) {

if (cadena[i] != cadena[j]) {

return 0; // No es un palíndromo

}

}

return 1;

}

int main() {

char texto[MAX\_LONGITUD];

printf("Ingrese un texto para verificar si es un palíndromo: ");

fgets(texto, sizeof(texto), stdin);

size\_t longitud = strlen(texto);

if (longitud > 0 && texto[longitud - 1] == '\n') {

texto[longitud - 1] = '\0';

}

for (int i = 0; i < longitud; i++) {

texto[i] = tolower(texto[i]);

}

eliminarEspacios(texto);

if (esPalindromo(texto)) {

printf("El texto ingresado es un palíndromo.\n");

} else {

printf("El texto ingresado no es un palíndromo.\n");

}

return 0;

}

13)

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#define MAX\_LONGITUD 100

int main() {

char texto[MAX\_LONGITUD];

printf("Ingrese un texto para invertir las letras: ");

fgets(texto, sizeof(texto), stdin);

size\_t longitud = strlen(texto);

if (longitud > 0 && texto[longitud - 1] == '\n') {

texto[longitud - 1] = '\0';

longitud--;

}

for (int i = 0; i < longitud; i++) {

if (isalpha(texto[i])) {

if (isupper(texto[i])) {

texto[i] = tolower(texto[i]);

} else if (islower(texto[i])) {

texto[i] = toupper(texto[i]);

}

}

}

printf("Texto con letras invertidas: %s\n", texto);

return 0;

}

14)

#include <stdio.h>

#include <ctype.h>

#define NUM\_ALUMNOS 3

#define NUM\_MATERIAS 5

void corregirMayuscula(char apellido[]) {

if (!isupper(apellido[0])) {

apellido[0] = toupper(apellido[0]);

}

}

float calcularPromedio(float notas[]) {

float suma = 0;

for (int i = 0; i < NUM\_MATERIAS; i++) {

suma += notas[i];

}

return suma / NUM\_MATERIAS;

}

int main() {

char apellidos[NUM\_ALUMNOS][50];

float notas[NUM\_ALUMNOS][NUM\_MATERIAS];

for (int i = 0; i < NUM\_ALUMNOS; i++) {

printf("Ingrese el apellido del alumno %d: ", i + 1);

scanf("%s", apellidos[i]);

corregirMayuscula(apellidos[i]);

printf("Ingrese las notas de las 5 materias para el alumno %d:\n", i + 1);

for (int j = 0; j < NUM\_MATERIAS; j++) {

printf("Nota de materia %d: ", j + 1);

scanf("%f", &notas[i][j]);

}

}

for (int i = 0; i < NUM\_ALUMNOS; i++) {

printf("\nAlumno %d - Apellido: %s\n", i + 1, apellidos[i]);

printf("Promedio: %.2f\n", calcularPromedio(notas[i]));

}

float promedioCurso = 0;

for (int i = 0; i < NUM\_ALUMNOS; i++) {

promedioCurso += calcularPromedio(notas[i]);

}

promedioCurso /= NUM\_ALUMNOS;

printf("\nPromedio del curso: %.2f\n", promedioCurso);

return 0;

}

15)

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#define MAX\_LONGITUD 100

int esVocal(char c) {

c = tolower(c);

return (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

}

int main() {

char texto[MAX\_LONGITUD];

printf("Ingrese un texto: ");

fgets(texto, sizeof(texto), stdin);

size\_t longitud = strlen(texto);

if (longitud > 0 && texto[longitud - 1] == '\n') {

texto[longitud - 1] = '\0';

}

int contadorA = 0, contadorE = 0, contadorI = 0, contadorO = 0, contadorU = 0;

for (int i = 0; i < longitud; i++) {

if (esVocal(texto[i])) {

switch (tolower(texto[i])) {

case 'a': contadorA++; break;

case 'e': contadorE++; break;

case 'i': contadorI++; break;

case 'o': contadorO++; break;

case 'u': contadorU++; break;

}

}

}

printf("\nCantidad de veces que se repite cada vocal:\n");

printf("A: %d\n", contadorA);

printf("E: %d\n", contadorE);

printf("I: %d\n", contadorI);

printf("O: %d\n", contadorO);

printf("U: %d\n", contadorU);

return 0;

}

16)

#include <stdio.h>

#include <string.h>

#define MAX\_LONGITUD\_NOMBRE 50

struct Empleado {

char nombre[MAX\_LONGITUD\_NOMBRE];

char fechaIngreso[11];

float sueldo;

};

int compararFechas(const char fecha1[], const char fecha2[]) {

int dia1, mes1, anio1;

int dia2, mes2, anio2;

sscanf(fecha1, "%d/%d/%d", &dia1, &mes1, &anio1);

sscanf(fecha2, "%d/%d/%d", &dia2, &mes2, &anio2);

if (anio1 < anio2) {

return -1;

} else if (anio1 > anio2) {

return 1;

} else {

if (mes1 < mes2) {

return -1;

} else if (mes1 > mes2) {

return 1;

} else {

if (dia1 < dia2) {

return -1;

} else if (dia1 > dia2) {

return 1;

} else {

return 0;

}

}

}

}

int main() {

struct Empleado empleados[3];

for (int i = 0; i < 3; i++) {

printf("Ingrese nombre del empleado %d: ", i + 1);

scanf("%s", empleados[i].nombre);

printf("Ingrese fecha de ingreso del empleado %d (formato dd/mm/yyyy): ", i + 1);

scanf("%s", empleados[i].fechaIngreso);

printf("Ingrese sueldo del empleado %d: ", i + 1);

scanf("%f", &empleados[i].sueldo);

}

int indiceEmpleadoMasAntiguo = 0;

for (int i = 1; i < 3; i++) {

if (compararFechas(empleados[i].fechaIngreso, empleados[indiceEmpleadoMasAntiguo].fechaIngreso) < 0) {

indiceEmpleadoMasAntiguo = i;

}

}

printf("\nEmpleado más antiguo:\n");

printf("Nombre: %s\n", empleados[indiceEmpleadoMasAntiguo].nombre);

printf("Sueldo: %.2f\n", empleados[indiceEmpleadoMasAntiguo].sueldo);

return 0;

}

17)

#include <stdio.h>

#include <string.h>

#include <time.h>

#define MAX\_LONGITUD\_NOMBRE 50

struct Persona {

char nombre[MAX\_LONGITUD\_NOMBRE];

int dia, mes, anio;

};

void obtenerFechaActual(int \*dia, int \*mes, int \*anio) {

time\_t tiempo;

struct tm \*tiempoLocal;

tiempo = time(NULL);

tiempoLocal = localtime(&tiempo);

\*dia = tiempoLocal->tm\_mday;

\*mes = tiempoLocal->tm\_mon + 1;

\*anio = tiempoLocal->tm\_year + 1900;

}

void cargarFechaNacimiento(struct Persona \*persona) {

int diaActual, mesActual, anioActual;

obtenerFechaActual(&diaActual, &mesActual, &anioActual);

do {

printf("Ingrese la fecha de nacimiento de %s (formato dd mm yyyy): ", persona->nombre);

scanf("%d %d %d", &persona->dia, &persona->mes, &persona->anio);

if (persona->anio > anioActual ||

(persona->anio == anioActual && persona->mes > mesActual) ||

(persona->anio == anioActual && persona->mes == mesActual && persona->dia > diaActual)) {

printf("Error: La fecha de nacimiento no puede ser posterior a la fecha actual. Intente nuevamente.\n");

}

} while (persona->anio > anioActual ||

(persona->anio == anioActual && persona->mes > mesActual) ||

(persona->anio == anioActual && persona->mes == mesActual && persona->dia > diaActual));

}

int calcularEdad(struct Persona \*persona) {

int diaActual, mesActual, anioActual;

obtenerFechaActual(&diaActual, &mesActual, &anioActual);

int edad = anioActual - persona->anio;

if (mesActual < persona->mes || (mesActual == persona->mes && diaActual < persona->dia)) {

edad--;

}

return edad;

}

int main() {

struct Persona personas[3];

for (int i = 0; i < 3; i++) {

printf("Ingrese el nombre de la persona %d: ", i + 1);

scanf("%s", personas[i].nombre);

cargarFechaNacimiento(&personas[i]);

}

char nombreModificar[MAX\_LONGITUD\_NOMBRE];

printf("\nIngrese el nombre de la persona para modificar la fecha: ");

scanf("%s", nombreModificar);

int indiceModificar = -1;

for (int i = 0; i < 3; i++) {

if (strcmp(personas[i].nombre, nombreModificar) == 0) {

indiceModificar = i;

break;

}

}

if (indiceModificar != -1) {

printf("Modificación de la fecha de nacimiento para %s:\n", nombreModificar);

cargarFechaNacimiento(&personas[indiceModificar]);

} else {

printf("No se encontró a una persona con el nombre ingresado.\n");

}

printf("\nDatos ingresados ordenados por edad:\n");

for (int i = 0; i < 3; i++) {

printf("Nombre: %s, Fecha de Nacimiento: %02d/%02d/%04d, Edad: %d años\n",

personas[i].nombre, personas[i].dia, personas[i].mes, personas[i].anio,

calcularEdad(&personas[i]));

}

return 0;

}