

# PRÁCTICA 1

## PROGRAMACIÓN 2

1. Dados los siguientes tipos de datos del lenguaje Pascal: *char*, *integer*, *longint*, *word*, *real* y *double*; obtener el tipo equivalente en el lenguaje C especificando la cantidad de bytes que ocupa y su cadena de formato.

Pascal	C	Bytes	Cadena Formato
char	char	1	%c
integer	int	4	%d
longint	long int	8	%ld
word	unsigned short	2	%hu
real	float	4	%f
double	double	8	%lf

2. Responder:

- a) ¿Cuál de los siguientes es ejemplo de un archivo de cabecera?: `printf`, `include`, `conio.h`, `#define`.
- b) ¿Cuál es la librería que se requiere para utilizar cada una de las siguientes funciones?: `printf`, `scanf`, `getch`, `getchar`.

- a) `conio.h`, un archivo de cabecera se distingue por llevar la terminación **.h**

- b)

Funcion	Libreria	Uso
<code>printf</code>	<code>stdio.h</code>	Imprime
<code>scanf</code>	<code>stdio.h</code>	Lee datos de entrada
<code>getch</code>	<code>conio.h</code>	Lee un carácter directamente del teclado sin mostrarlo en la consola. No requiere que se presione Enter.
<code>getchar</code>	<code>stdio.h</code>	Lee un carácter y lo devuelve como un valor entero.

3. Describir lo que imprimen los siguientes fragmentos de código (justificar si no imprimen nada):

a) `char A;`  
`A = 'a';`  
`printf(" %c ", A);` \_\_\_\_\_  
`printf(" %d ", A);` \_\_\_\_\_  
`A = 'a' + 10;` \_\_\_\_\_  
`printf(" %c ", A);` \_\_\_\_\_  
`printf(" %d ", A);` \_\_\_\_\_

b) `int B;`  
`B = 5;`  
`printf(" %d ", B);` \_\_\_\_\_  
`printf(" %c ", B);` \_\_\_\_\_  
`B = 5 + 'A';` \_\_\_\_\_  
`printf(" %d ", B);` \_\_\_\_\_  
`printf(" %c ", B);` \_\_\_\_\_

c) `#define num 15`  
`printf(" %d ", num);` \_\_\_\_\_  
`printf(" %c ", num);` \_\_\_\_\_

d) `#define num 15`  
`num = 5 + 'A';`  
`printf(" %d ", num);` \_\_\_\_\_  
`printf(" %c ", num);` \_\_\_\_\_

Codigo	Consola
<code>printf(" %c ", A);</code>	a
<code>printf(" %d ", A);</code>	97
<code>printf(" %c ", A);</code>	k
<code>printf(" %d ", A);</code>	107
<code>printf(" %d ", B);</code>	5
<code>printf(" %c ", B);</code>	Caracter no imprimible
<code>printf(" %d ", B);</code>	70
<code>printf(" %c ", B);</code>	F
<code>printf(" %d ", num);</code>	Error de compilacion en <code>num = 5 + 'A';</code>
<code>printf(" %c ", num);</code>	Error de compilacion en <code>num = 5 + 'A';</code>
<code>printf(" %d ", num);</code>	15
<code>printf(" %c ", num);</code>	Carácter no imprimible

4. Completar el resultado que muestran los siguientes programas:

```
a) #include <stdio.h>
int main(){
    int a, b, c = 3, d = 2;
    a = 8 - 5 * 3 + 2;
    b = 7 % 3 + 4 * 2;
    printf(" valor de a %d\tvalor de b %d\n", a ,b); _____
    b %= a;
    printf(" valor de b %d\t\n", b); _____
    b = (-4) % 3;
    printf(" valor de b %d\t\n", b); _____
    if(d = a)
        printf(" d y a son iguales ");
    else printf(" d y a son distintos ");
    return 0;
}

b) #include <stdio.h>
int main(){
    int c=3, d=2;
    c -= d++ * 3;
    printf(" valor de c %d\tvalor de d %d\n", c, d); _____
    return 0;
}

c) #include <stdio.h>
int main() {
    int c = 3, l;
    c = (c * 2 - (l = 4, -- l));
    printf(" valor de c %d\t valor de l %d\n", c, l); _____
    getch();
    return 0;
}
```

Codigo	Resultado
printf(" valor de a %d\tvalor de b %d\n", a ,b);	valor de a -5 valor de b 9
printf(" valor de b %d\t\n", b);	valor de b 4
printf(" valor de b %d\t\n", b);	valor de b -1
if(d = a) printf(" d y a son iguales "); else printf(" d y a son distintos ");	d y a son iguales
printf(" valor de c %d\tvalor de d %d\n", c, d);	valor de c -3 valor de d 3
printf(" valor de c %d\t valor de l %d\n", c, l);	valor de c 3 valor de l 3

5. Convertir las siguientes expresiones lógicas en Pascal al lenguaje C:

- a) ( z < 5 ) AND ( ( a < = 5 ) OR ( c > 0 ) )  
b) ( z = CHR( 0 ) ) OR NOT( total = 0 )

- a) (z < 5) && ((a < = ) || (c > 0))  
b) (z == "\0") || !(total == 0)

6. Obtener el resultado de las siguientes expresiones, previa asignación de valores a las cuatro variables:

- a) a + b \* c / d  
b) (a + b) \* c % d  
c) d < 65 && 3 == d  
d) a <= b || (3 > a)  
e) (a >= d) && (2 - 14 == c + 1)  
f) a>b || a<c && a = c || a >= b

- a) 8
- b) 0
- c) 0
- d) 1
- e) 0
- f) 0

7. Indicar los valores que almacenan las variables en las siguientes instrucciones de lectura:

<p>a) <code>int a;</code> <code>/* leer a = 3 */</code> <code>scanf("%d", &amp;a);</code> _____ <code>scanf("%f", &amp;a);</code> _____ <code>scanf("%c", &amp;a);</code> _____</p> <p>b) <code>char x;</code> <code>/* leer x = 3 */</code> <code>scanf("%d", &amp;x);</code> _____ <code>scanf("%f", &amp;x);</code> _____ <code>scanf("%c", &amp;x);</code> _____</p>	<p>c) <code>char w;</code> <code>/* leer w = 2.5 */</code> <code>scanf("%d", &amp;w);</code> _____ <code>scanf("%f", &amp;w);</code> _____ <code>scanf("%c", &amp;w);</code> _____</p>
--	--

Para `int a`, solo `scanf("%d", &a);` es correcto.

Para `char x`, solo `scanf("%c", &x);` es correcto.

Para `char w`, solo `scanf("%c", &w);` es correcto.

8. Completar la expresión del siguiente *if* para que se ejecute su parte verdadera y la variable `a` contenga el valor 1 luego de esa sentencia.

```
int a = 0;
if(    )
    printf("a vale 0");
else
    printf("a vale 1");
```

```
int a = 0;
if(!a) {
    printf("a vale 0");
    a = 1;
} else {
    printf("a vale 1");
}
```

9. Reescribir las siguientes sentencias utilizando la instrucción `switch`:

```
If (x == 4)
    y = 7;
else
    if (x == 5)
        y = 9;
    else
        if (x == 9)
            y = 14;
        else
            y = 22;
```

```
switch(x){
    case 4:
        y = 7;
        break;
    case 5:
        y = 9;
```

```
        break;
case 9:
    y = 14;
    break;
default:
    y = 22;
    break;
}
```

10. Reescribir cada una de las siguientes porciones de código en una sola sentencia mediante el uso del operador condicional:

a) `if (a < b)`  
    `c = a;`  
    `else c = b;`

b) `if (a < b && b < c)`  
    `printf("ordenados");`  
    `else printf("desordenados");`

- a) `c = (a < b) ? a : b;`  
b) `printf("%s", (a < b && b < c) ? "ordenados" : "desordenados");`

11. Desarrollar una macro que devuelva cada uno de los siguientes resultados:

a) Máximo de 2 números

b) Caracter es alfabético

Implementar un programa que utilice las macros desarrolladas.

- a) `#define max(a,b) (a < b) ? a : b;`  
b) `#define alfabetical(c) ((c > 'A' && c < 'Z') || (c > 'a' && c < 'z')) ? "alfabetico" : "no alfabetico";`  
`void main ()`  
    `int n1,n2;`  
    `char c;`  
    `scanf("%d %d %c", &n1,&n2,&c);`  
    `printf("el maximo es: %d",max(n1,n2));`  
    `printf("%c %s",c, alfabetical(c));`  
`}`

12. Desarrollar un programa que muestre la suma de un conjunto de números naturales ingresados por teclado. La serie finaliza al ingresar el número 0.

```
void main () {
    int num,suma = 0;
    do {
        scanf("%d",&num);
        suma += num;
    } while (num != 0);
    printf("%d", suma);
}
```

13. Corregir los errores del siguiente fragmento de código para que muestre los números naturales impares hasta el 15 inclusive.

```
#define <stdio.h>
#define N 9

int i;
for(i = 1; I < N; i++){
    printf("%d", &I);
```

```
#define <stdio.h>
#define N 15
void main(){
    int i;
    for(i = 1; i < N; i += 2){
        printf("%d", i);
    }
}
```

14. Desarrollar un subprograma que muestre un cuadrado con bordes de N numerales (N es parámetro de entrada). Por ej: para N = 4

```
####
#  #
#  #
#  #
####
```

```
void main (){
    int n,i,j;
    scanf("%d", &n);

    //tope
    for (i := 0; i < n; i++){
        printf("#");
    }
    printf("\n");

    //interior
    for (i := 1; i < n - 1; i++){
        printf("#");
        for (j := 1 ; j < n - 1; j ++){
            printf(" ");
        }
        printf("#\n");
    }

    //tope
    for (i := 0; i < n; i++){
        printf("#");
    }
}
```

15. Desarrollar una función que reciba tres números enteros y devuelva si son consecutivos entre sí o si no lo son.

```
int consecutivos (int n1,int n2,int n3){
```

```
int aux;
if (n1 > n2) {
    aux = n1;
    n1 = n2;
    n2 = aux;
}
if (n2 > n3) {
    aux = n2;
    n2 = n3;
    n3 = aux;
}
if (n1 > n2) {
    aux = n1;
    n1 = n2;
    n2 = aux;
}
return (n1 + 1 == n2) && (n2 + 1 == n3);
}

void main () {
    int n1,n2,n3;
    scanf("%d %d %d",&n1,&n2,&n3);
    printf("%s",consecutivos(n1,n2,n3) ? "consecutivos" : "no consecutivos");
}
```

16. Desarrollar un subprograma que reciba una cantidad de segundos y devuelva su equivalente en el formato *horas:minutos:segundos*

```
void tiempo (int segx, int *hr, int *min, int *seg){
    *hr = segx /3600;
    segx %= 3600;
    *min = segx / 60;
    *seg = segx % 60;
}

void main (){
    int segx,hr,min,seg;
    scanf("%d",&segx);
    tiempo(segx,&hr,&min,&seg);
    printf("%d:%d:%d",hr,min,seg);
}
```

17. Enumerar los valores de todos los componentes de los siguientes arreglos. Especificar los arreglos de caracteres que pueden ser utilizados correctamente como cadenas.

- |                        |                                    |
|------------------------|------------------------------------|
| a) int v1[4] = {0};    | d) char s1[4] = {'h','o','y'};     |
| b) int v2[5] = {6};    | e) char s2[] = {'h','o','y'};      |
| c) int v3[] = {2,4,6}; | f) char s3[4]= {'h','o','y','\0'}; |

a) vector de 0s y basura en la cuarta posicion

- b) vector de 0s y 6 en la primera posición
- c) vector con 2 4 6 en las primeras 3 posiciones, el resto basura
- d) almacena la palabra "hoy" se puede usar como cadena ya que destina el elemento sobrante como \0
- e) almacena la palabra "hoy" seguida de caracteres basura, no se puede usar como cadena
- f) almacena la palabra "hoy" se puede usar como cadena

18. Desarrollar un subprograma que duplique los elementos de un vector ubicados en las posiciones pares. Implementar 3 veces utilizando *for*, *while* y *do while*

## for

```
void duplicar_pares (int V[],int n){
    int i;
    for (i = 1; i < n; i += 2){
        V[i] *= 2;
    }
}
```

## while

```
void duplicar_pares (int V[],int n){
    int i = 1;
    while (i < n){
        V[i] *= 2;
        i += 2;
    }
}
```

## do while

```
void duplicar_pares (int V[],int n){
    int i = 1;
    do{
        V[i] *= 2;
        i += 2;
    } while (i < n);
}
```

19. Responder, siendo x un arreglo de 10 enteros (int x[10]):

- a) ¿Se puede determinar por la siguiente invocación `funarr(x, n)` si x es parámetro de entrada o de salida?
  - b) ¿Se puede acceder al contenido de `x[10]`? ¿Qué representa el valor obtenido?
- 
- a) No, ya que los vectores ya son un vector de punteros y no se debe agregar & para indicar el espacio en memoria.
  - b) El vector es de 10 elementos, pero el primer elemento es el 0 y el último el 9 por lo que `x[10]` no representa un entero almacenado. El resultado es indefinido.



20. Realizar un programa iterativo que calcule e imprima el equivalente binario de un número entero positivo dado.

```
#include <stdio.h>
#define MAX 30

void binario (int num, int v[], int *n){
    *n = 0;
    while (num >= 1) {
        v[*n] = num % 2;
        num /= 2;
        (*n)++;
    }
}

int main (){
    int n,num,v[MAX];
    printf("Introduce un número: ");
    scanf("%d",&num);
    binario(num,v,&n);
    for (int i = n-1; i >= 0; i--){
        printf("%d",v[i]);
    }
    return 0;
}
```

21. Desarrollar un programa que lea un vector y, mediante opciones de un menú, calcule e imprima lo siguiente:

- c) los elementos en posiciones impares 1,3,5 o sea segundo, cuarto, sexto...
- d) genere otro vector con los elementos divisibles por k (k dato). Muestre el nuevo vector
- e) busca la posición de un valor x (puede no existir)

```
#include <stdio.h>
#define MAX 30

void pos_impar (int n, int v[]){
    for (int i = 0; i < n; i += 2){
        printf("%d",v[i]);
    }
}

void divisibles (int v[], int div[], int k, int n){
    int m = 0;
    for (int i = 0; i < n; i++){
        if (v[i] % k == 0) {
            div[m] = v[i];
            m++;
        }
    }
    for (int i = 0; i < m; i++) {
        printf("%d ", div[i]);
    }
}
```

```
}
}
int posicionx (int v[], int n, int x){
int i = 0;
while ((i < n) && (x != v[i]))
    i++;
if (i < n) {
    return i;
} else {
    return -1;
}
}
int main (){
int n,num, div[MAX],v[MAX],rta,k,x, pos;
printf("Introduce el tamaño del vector: ");
scanf("%d", &n);
printf("Introduce los elementos del vector: ");
for (int i = 0; i < n; i++) {
    scanf("%d", &v[i]);
}
printf("Introduce una opcion (1: mostrar pos impares\t2:generar vector divisibles\t3:posicion
x): ");
scanf("%d",&rta);
switch (rta){
    case 1:
        pos_impar(n,v);
        break;
    case 2:
        printf("ingrese divisor: ");
        scanf("%d",&k);
        divisibles(v,div,k,n);
        break;
    case 3:
        printf("ingrese entero: ");
        scanf("%d",&x);
        pos = 1 + posicionx(v,n,x);
        if (pos >= 0) {
            printf("Pos %d\n", pos);
        } else {
            printf("No existe\n");
        }
        break;
    default:
        printf("opcion no valida");
        break;
}
return 0;
}
```

22. Responder:

- a) ¿Por qué no es correcta la siguiente sentencia: `scanf("%s", &cad)?`
  - b) ¿Qué instrucción resulta conveniente para leer por teclado apellido y nombre de una persona: `gets` o `scanf`? Justificar la elección.
  - c) ¿Qué representan los valores devueltos por la función `strcmp`?
- a) no es correcta porque cad ya es un puntero a la primera posición del arreglo de caracteres. Al usar el operador de dirección & se está pasando un puntero a un puntero, lo cual no es lo que se espera para la función scanf con el especificador de formato %s.
  - b) gets() es la indicada para leer cadenas de caracteres ya que maneja mejor los espacios
  - c) representa valores entre -1 y 1 para comparar dos cadenas de caracteres siendo 0 que ambas cadenas son iguales

23. Corregir el siguiente código para que asigne el contenido de la cadena cad1 a cad3, solamente si cad1 y cad2 son distintas.

```
char cad1[20], cad2[20], cad3[20];
gets(cad1); gets(cad2);
if(cad1 != cad2)
    cad3 = cad1;
```

```
char cad1[20], cad2[20], cad3[20];
gets(cad1); gets(cad2);
if(strcmp(cad1,cad2) != 0)
    strcpy(cad3 , cad1);
```

24. Desarrollar un programa que lea una matriz de enteros de NxM y:

- a) busque la posición de un valor x (puede no existir)
- b) muestre el promedio de cada columna de la matriz.
- c) muestre para cada fila que tenga al menos un cero la cantidad de elementos que superan el promedio de la matriz

```
#include <stdio.h>
#define MAX 30
```

```
void posicion (int *x,int *y,int n,int m,int num,int mat[][m]){
    int i,j;
    i = 0;j = 0;
    while(i < n && mat[i][j] != num){
        if (j < m){
            j++;
        } else {
            j = 0;
            i++;
        }
    }
    if (i < n){
        *x = i;
        *y = j;
    }
}
```

```
    } else {
        *x = -1;
        *y = -1;
    }
}

void mostrar_promedios (int n, int m, int mat[][m]){
    int i,j;
    float suma,prom;
    for (i = 0; i < n; i++) {
        suma = 0;
        for (j = 0; j < m; j++){
            suma += mat[i][j];
        }
        prom = suma/m;
        printf("promedio de la columna: %f",prom);
    }
}

float promedio (int n,int m, int mat[][m]){
    int i,j;
    float suma = 0;
    for (i = 0; i < n; i++){
        for (j = 0; j < m; j++){
            suma += mat[i][j];
        }
    }
    return suma/(n*m);
}

void supera_promedio (int n,int m, int mat[][m]){
    float prom = promedio(n,m,mat);
    int cumple,cont,i,j;
    for (i = 0; i < n; i++){
        cumple = 0;
        cont = 0;
        for (j = 0; j < m; j++){
            if (cont > prom)
                cont++;
            if (mat[i][j] == 0)
                cumple = 1;
        }
        if(cumple)
            printf("Fila %d supera el promedio: %d elementos\n", i, cont);
    }
}

int main () {
    int num,x,y,n,m, mat[MAX][MAX];
    char rta;
    printf("ingrese una opcion a) posición de un valor b) promedio de cada columna c) mayores al promedio: ");
    scanf("%c",&rta);
}
```

```
switch(rta){
    case 'a':
        printf("ingrese un valor: ");
        scanf("%d",&num);
        posicion(&x,&y,n,m,num,mat);
        break;
    case 'b':
        mostrar_promedios(n, m, mat);
        break;
    case 'c':
        supera_promedio(n, m, mat);
        break;
    default:
        printf("opcion invalida");
        break;
}
return 0;
}
```

25. Leer el nombre y la edad de N personas, calcular e imprimir:

- a) el listado de nombres de las personas de más de X años (X es dato).
- b) el porcentaje de las personas mayores de edad (sobre el total).

```
#include <stdio.h>
#define MAX 30

typedef struct {
    int edad;
    char nombre[20];
}regpersona;

void cargar_estructura(int *n,regpersona persona[]){
    int i;
    printf("ingrese cantidad personas: ");
    scanf("%d",n);
    for (i = 0; i < *n; i++){
        scanf("%d %s",&persona[i].edad,persona[i].nombre);
    }
}

void listar_y_porcentaje (int x, regpersona persona[],int n){
    int i,conttot = 0;
    float contmayor = 0;
    for(i = 0; i < n; i++){
        conttot++;
        if (persona[i].edad > x){
            printf("%s %d\n", persona[i].nombre, persona[i].edad);
        }
        if (persona[i].edad >= 18){
            contmayor++;
        }
    }
}
```

```
}  
printf("%5.2f % mayor de edad", 100*contmayor/conttot);  
}  
  
int main (){  
    regpersona persona[MAX];  
    int n,x;  
    cargar_estructura(&n,persona);  
    printf("ingrese edad minima: ");  
    scanf("%d",&x);  
    listar_y_porcentaje(x,persona,n);  
    return 0;  
}
```

26. Declarar un tipo struct para almacenar nombre, matrícula y promedio de un alumno.  
Desarrollar una función que realice lectura de una variable del tipo creado.

```
typedef struct {  
    char nombre[30];  
    char matricula[8];  
    float promedio;  
}regalum;  
  
void leer_struct (regalum alumno){  
    printf("Ingrese matricula: ");  
    scanf("%s", alumno.matricula);  
    printf("Ingrese nombre: ");  
    scanf("%s", alumno.nombre);  
    printf("Ingrese promedio: ");  
    scanf("%f", &alumno.promedio);  
}
```

27. Desarrollar una función que a partir de una matriz de NxM elementos enteros, devuelva en un arreglo de structs, fila, columna y valor de las componentes de la matriz divisibles por N+M

```
#include <stdio.h>  
  
#define MAX 30  
  
typedef struct {  
    int fila;  
    int columna;  
    int valor;  
}regmat;
```

```
void armar_arreglo (int *c, int mat[][m], regmat divisibles[]){
    int i,j;
    for (i = 0; i < n; i++){
        for(j = 0; j < m; j++){
            if (mat[i][j] % (n+m) == 0){
                divisibles[*c].valor = mat[i][j];
                divisibles[*c].columna = j;
                divisibles[*c].fila = i;
                (*c)++;
            }
        }
    }
}

int main(){
    int c = 0,n,m, mat[MAX][MAX];
    regmat divisibles[MAX];
    cargarmatriz(&n,&m,mat);
    armar_arreglo(&c,mat,divisibles);
}
```

28. Rehacer el ejercicio 12 suponiendo que los números se encuentran en un archivo de texto, todos en la misma línea. ¿Como modificaría la lectura, si hubiera un valor por línea?

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

void main(){
    FILE *arch;
    int num,suma = 0;
    arch = fopen("numeros.txt","r");
    if (arch == NULL){
        printf("archivo inexistente");
    } else {
        while(!feof(arch)){
            fscanf(arch,"%d",&num);
            suma += num;
        }
        fclose(arch);
        printf("%d",suma);
    }
}
```

29. Un sensor de temperatura realiza mediciones periódicas (una cada 5 seg.). Las mismas son registradas en un archivo binario, donde cada elemento es de tipo *float*. Implementar un programa que genere dicho archivo y, otro programa que lo lea y muestre por pantalla temperatura promedio, máxima y mínima.

## Programa que genera el archivo

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

```
void main (){
    FILE *arch;
    float temp;
    arch = fopen("temperaturas.dat","wb");
    printf("ingrese la temperatura: ");
    scanf("%f",&temp);
    while (temp != 999){
        fwrite(&temp,sizeof(float),1,arch);
        printf("ingrese la temperatura: (999 finaliza)");
        scanf("%f",&temp);
    }
    fclose(arch);
}
```

## Programa que calcula y muestra

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

void main (){
    FILE *arch;
    float temp,suma = 0,max = -999,min = 999,prom;
    int n = 0;
    arch = fopen("temperaturas.dat","rb");
    if (arch == NULL){
        printf("error al abrir archivo");
    } else {
        while (!feof(arch)){
            fread(&temp,sizeof(float),1,arch);
            suma +=temp;
            n++;
            if (temp > max)
                max = temp;
            if (temp < min)
                min = temp;
        }
        fclose(arch);
        prom = suma/n;
        printf("promedio: %5.2f\tmax: %5.2f\tmin: %5.2f",prom,max,min);
    }
}
```

30. Desarrollar un programa que lea y procese un archivo de texto generado por un radar vigía. Cada línea contiene patente de un vehículo, velocidad registrada, velocidad máxima permitida, fecha y hora de la medición. Cada dato está separado por un espacio. El programa debe generar un archivo binario con los registros donde la velocidad registrada supere en más de un 20% a la velocidad máxima. Cada registro del archivo binario contiene los mismos datos que el archivo de texto. Mostrar, al final del proceso, la cantidad de mediciones procesadas y la cantidad de registros generados en el archivo binario.

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



```
#include <stdlib.h>

typedef struct {
    float velregistrada;
    float velmax;
    char patente[8];
    char fecha[9];
    int hora;
} regmulta;

int main() {
    regmulta multa;
    float velregistrada, velmax;
    char patente[8], fecha[9];
    int hora, n = 0, m = 0;
    FILE *arch, *archb;
    arch = fopen("radar.txt", "r");
    archb = fopen("infracciones.dat", "wb");
    if (arch == NULL || archb == NULL) {
        printf("Error al abrir los archivos");
    } else {
        while (fscanf(arch, "%s %f %f %s %d", patente, &velregistrada, &velmax, fecha, &hora) != EOF)
        {
            n++;
            if (velregistrada >= velmax * 1.2) {
                strcpy(multa.patente, patente);
                strcpy(multa.fecha, fecha);
                multa.hora = hora;
                multa.velregistrada = velregistrada;
                multa.velmax = velmax;
                fwrite(&multa, sizeof(regmulta), 1, archb);
                m++;
            }
        }
    }
    printf("Se procesaron %d mediciones.\nSe registraron %d infracciones.\n", n, m);
    fclose(arch);
    fclose(archb);

    return 0;
}
```

**31. El ranking de una asociación de tenistas es registrada en un archivo binario. Para cada jugador se registra apellido y nombre, nacionalidad y puntaje. Realizar un subprograma que muestre los datos del jugador ubicado en una posición recibida como parámetro.**

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

```
typedef struct {
    char nombre[30];
    char apellido[30];
    char nacionalidad[30];
    float puntaje;
}TReg;

void posicionX (int x){
    FILE *arch;
    TReg reg;
    arch = fopen("ranking.dat","rb");
    if (arch == NULL) {
        printf("archivo inexistente");
    } else {
        fseek(arch,(x-1) * sizeof(TReg),SEEK_SET);
        fread(&reg,sizeof(TReg),1,arch);
        printf("nombre: %s\n apellido: %s\n nacionalidad: %s\n puntaje: %.2f",
reg.nombre,reg.apellido,reg.nacionalidad,reg.puntaje);
    }
    fclose(arch);
}

int main(){
    int x;
    printf("ingrese posicion: ");
    scanf("%d",&x);
    posicionX(x);
    return 0;
}
```

32. En un archivo binario se registran temperaturas diarias máxima y mínima para algunas ciudades del país en un mes determinado. Cada registro tiene los siguientes campos: ciudad (cadena de 20, archivo ordenado por este campo), día del mes (1..30), temperatura máxima (float), temperatura mínima (float). Listar para cada ciudad sus temperaturas máximas y mínimas en el mes; y al final del listado la ciudad que tuvo la menor temperatura en el mes, indicando el día.

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

typedef struct {
    char ciudad[30];
    int dia;
    float temp_max;
    float temp_min;
```

```
} TReg;

int main() {
    FILE *arch;
    TReg reg;
    char ciudad_min[30], actciudad[30];
    int dia_min;
    float max,min,min_mes;

    archivo = fopen("temperaturas.dat", "rb");
    if (archivo == NULL) {
        printf("Error al abrir el archivo.\n");
    } else {
        fread(&reg, sizeof(TReg), 1, arch);
        strcpy(actciudad,reg.ciudad);
        max = reg.temp_max; min = reg.temp_min; min_mes = reg.temp_min;
        do{
            if (strcmp(actciudad, reg.ciudad) != 0) {
                printf("Ciudad: %s\n", actciudad);
                printf("Temperatura Máxima: %.2f\n", maxtemp);
                printf("Temperatura Mínima: %.2f\n\n", mintemp);
                strcpy(actciudad, reg.ciudad);
                maxtemp = reg.temp_max; mintemp = reg.temp_min;
            } else {
                if (reg.temp_max > maxtemp) {
                    maxtemp = reg.temp_max;
                }
                if (reg.temp_min < mintemp) {
                    mintemp = reg.temp_min;
                }
                if (reg.temp_min < min_mes) {
                    min_mes = reg.temp_min;
                    dia_min = reg.dia;
                    strcpy(ciudad_min, reg.ciudad);
                }
            }
        } while !feof(arch);
        printf("Ciudad: %s\n", actciudad);
        printf("Temperatura Máxima: %.2f\n", maxtemp);
        printf("Temperatura Mínima: %.2f\n\n", mintemp);
        printf("La ciudad con la menor temperatura del mes fue %s con %.2f°C el día %d.\n",
ciudad_min, min_mes, dia_min);
    }
    fclose(arch);
    return 0;
}
```