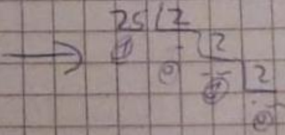
EJERCICIO MIXTO CON LOS ULTIMOS TEMAS VISTOS

LISTA 30 ALUMNOS CON LA ESTRUCTURA, CARGAR LOS 30 ALUMNOS, ORDENAR DNI + APELLIDO, BUSCAR UN ALUMNO POR DNI, MOSTRAR EL ALUMNO Q' ELIJAS Y PASAR EL DNI A BINARIO.

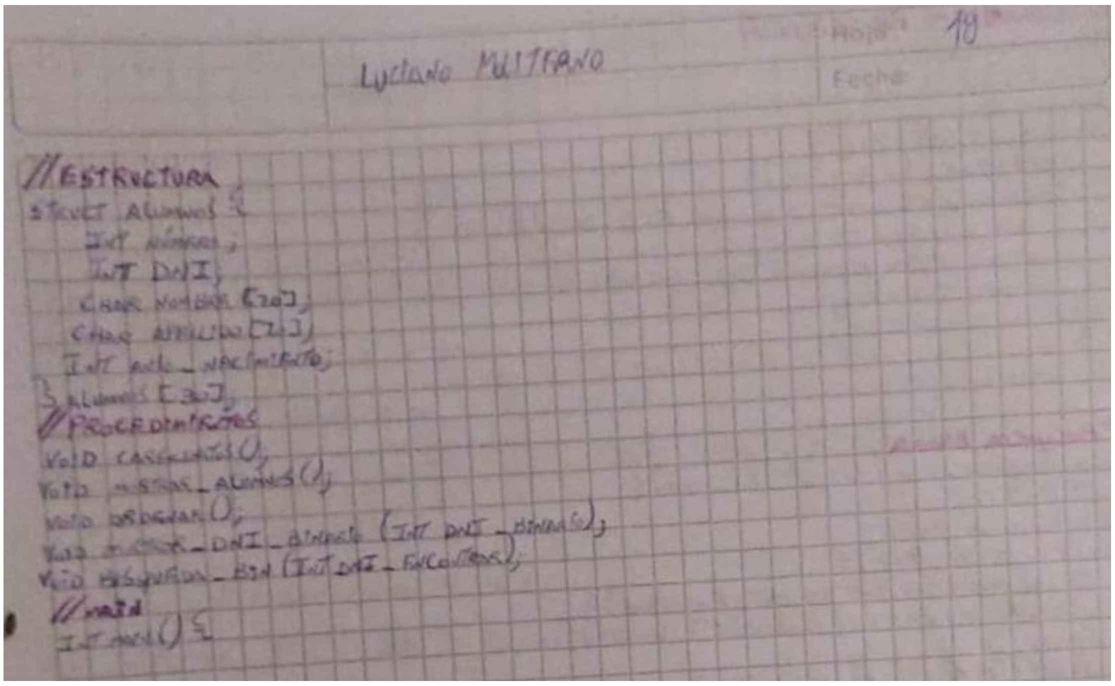
```
//ESTRUCTURA
  STRUCT ALUMNO {
        INT NÚMERO;
        INT DNI ;
       CHAR NOMBRE [20];
       CHAR APELLIDO [20];
      INT AÑO_NACIMIENTO;
    } ALUMNOS [30];
//FUNCIONES
INT BUSQUEDA_BIN (INT DNI_ENCONTRAR); VOID CARGADATOS ();
VOID MOSTRAR_ALUMNOS ();
VOID ORDENAR (INT CANTALUMNOS);
VOID MOSTRAR_DNI_EN_BINARIO (INT DNI TRANSFORMA);

// MAIN
  INT MAIN() {

    }
VOID MOSTRAR DNI EN BINARIO (INT DNI TRANSFORMA) {
      IF (DNI TRANSFORMA > 1) {
       // DNI TRANSFORMA = DNI TRANSFORMA/2;
        PRINTF ("%d", DNI TRANSFORMA %2); //GUARDAR EN UN VECTOR
        DNI TRANSFORMA = DNI TRANSFORMA/2;
       MOSTRAR DNI BINARIO (DNI TRANSFORMA) //RESTO EN RECURSIVIDAD
          //[illegible]
      }
```



RESTO ES BINARIO
CON RECURSIVIDAD

$I < \cancel{=} 30 = I < 30$ (EN LA FOTO)

USAR GETS EN VEZ DE SCANF (EN LA [illegible]) ⇒ DA STRINGS.
" FFLUSH EN CADA SCANF (" " ") ⇒ DEBAJO.



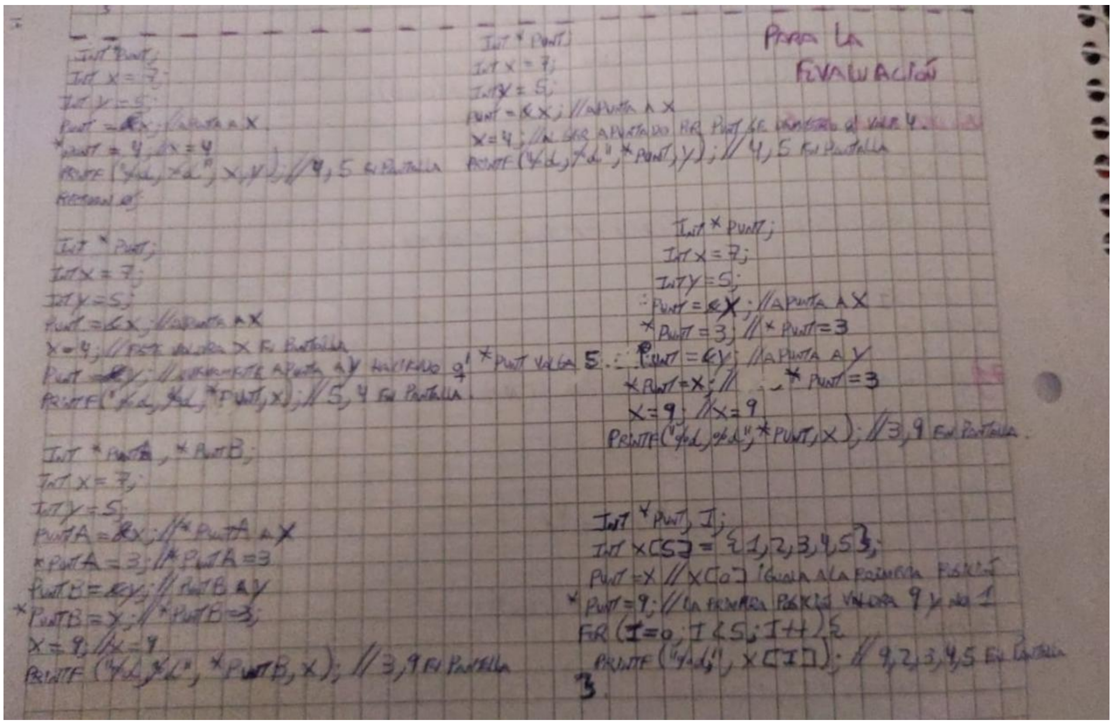
Hoja 19

Fecha

Luciano MULTFANO

```
//ESTRUCTURA
STRUCT ALUMNOS {
     INT NUMERO;
     INT DNI;
    CHAR NOMBRE [20];
   CHAR APELLIDO [20];
  INT AÑO_NACIMIENTO;
} ALUMNOS [30];
//PROCEDIMIENTOS
VOID CARGAR_DATOS ();
VOID MOSTRAR_ALUMNOS ();
VOID ORDENAR ();
VOID BUSCAR_DNI_BINARIO (INT DNI_BINARIO);
VOID BUSQUEDA_BIN (INT DNI_ENCONTRAR);
 //MAIN
 INT MAIN() {
```

```
//ORDENAMIENTO POR SELECCIÓN
VOID ORDENAR (){
 INT I,J;
ALUMNOS AUX;
FOR(I=0; I<29; I++){
  FOR(J=I+1; J<30; J++){
   IF(ALUMNOS [J].DNI > ALUMNOS [I].DNI){
     AUX = ALUMNOS [I];
     ALUMNOS [I]= ALUMNOS [J];
    ALUMNOS [J] = AUX;
   }
  }
 }
}
// BUSQUEDA BINARIA
VOID BUSQUEDA_BIN (INT DNI_ENCONTRAR){
   INT I=0;
BOOL ENCONTRADO= FALSE;
FOR(I=0, I<30, I++){
  IF(ALUMNOS [I]. DNI == DNI_ENCONTRAR){
    PRINTF("NOMBRE: %S", ALUMNOS [I]. NOMBRE);
   PRINTF("\n");
   PRINTF("APELLIDO: %S", ALUMNOS [I]. APELLIDO);
  PRINTF ("AÑO DE NACIMIENTO: %d", ALUMNOS [I]. AÑO_NACIMIENTO);
  PRINTF ("DNI: %d", ALUMNOS [I]. DNI);
 ENCONTRADO = TRUE;
 }
}
IF(ENCONTRADO == FALSE){
   PRINTF ("NO SE ENCONTRO EL DNI INGRESADO");
 }
//PASAJE ENTERO A BINARIO
VOID MOSTRAR_DNI_BINARIO (INT DNI_BINARIO){
 INT I, J, BINARIO [500], X;
 MOSTRAR_ALUMNOS ();
 FOR (I=0; I<30; I++){
  IF (ALUMNOS [I].DNI == DNI_BINARIO){
   FOR (I=500; I>0; I--){
      BINARIO [I] = X%2;
     X= X/2;
  }
   FOR (I=1; I<=500; I++){
     IF(BINARIO [I]==1){
     FOR (J=1; J<=500; J++){
       PRINTF ("%d", BINARIO [J]);
      }
      BREAK;
   }
  }
  SYSTEM ("PAUSE");
 }
}
}
```



PARA LA EVALUACIÓN

```
INT *PUNT;
INT X = 7;
INT Y = 5;
PUNT = &X; //APUNTA A X
*PUNT = 4; //X = 4
PRINTF ("%d, %d", X, Y); //4, 5 EN PANTALLA
RETURN 0;
```

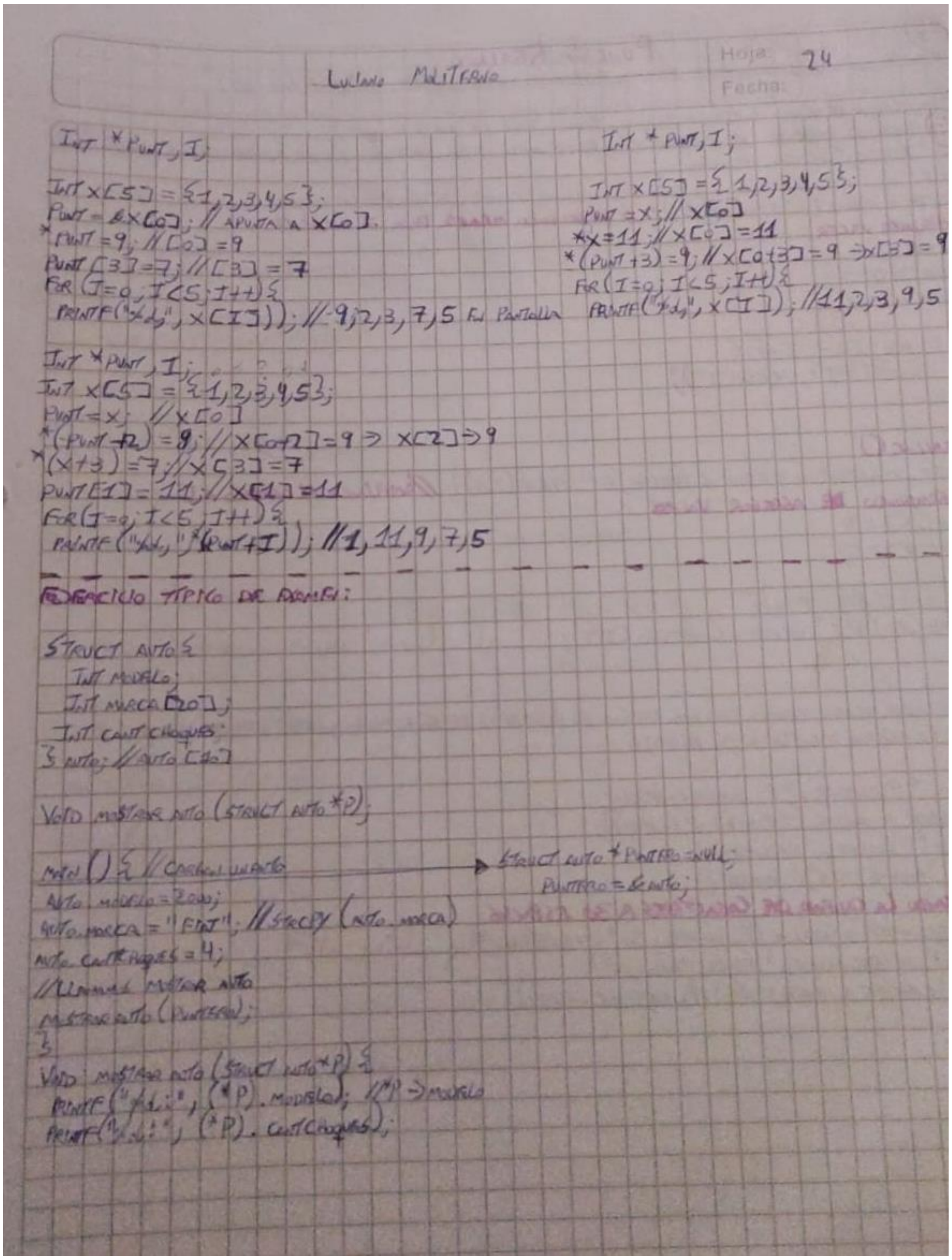
```
INT *PUNT;
INT X = 7;
INT Y = 5;
PUNT = &X; //APUNTA A X
X = 4; //AL SER APUNTADO POR PUNT SE [illegible] EL VALOR 4.
PRINTF ("%d, %d", *PUNT, Y); //4, 5 EN PANTALLA
```

```
INT *PUNT;
INT X = 7;
INT Y = 5;
PUNT = &X; //APUNTA A X
X = 4; //ESTE VALOR X EN PANTALLA
PUNT = &Y; //[illegible] APUNTA A Y HACIENDO Q' *PUNT VALGA 5.
PRINTF ("%d, %d", *PUNT, X); //5, 4 EN PANTALLA.
```

```
INT *PUNT;
INT X = 7;
INT Y = 5;
PUNT = &X; //APUNTA A X
*PUNT = 3; //*PUNT = 3
PUNT = &Y; //APUNTA A Y
*PUNT = X; //*PUNT = 3
X = 9; //X = 9
PRINTF ("%d, %d", *PUNT, X); //3, 9 EN PANTALLA.
```

```
INT *PUNTA, *PUNTB;
INT X = 7;
INT Y = 5;
PUNTA = &X; //*PUNTA A X
*PUNTA = 3; //*PUNTA = 3
PUNTB = &Y; //PUNTB A Y
*PUNTB = X; //*PUNTB = 3;
X = 9; //X = 9
PRINTF ("%d, %d", *PUNTB, X); //3, 9 EN PANTALLA
```

```
INT *PUNT, I;
INT X[5] = {1, 2, 3, 4, 5};
PUNT = X //X[0] IGUALA A LA PRIMERA POSICIÓN
*PUNT = 9; //LA PRIMERA POSICIÓN VALDRA 9 Y NO 1
FOR (I = 0; I < 5; I++){
  PRINTF ("%d", X[I]); //9, 2, 3, 4, 5 EN PANTALLA
}.
```



LUCIANO MALITERNO

Hoja: 24

Fecha:

```
INT *PUNT, I;

INT X[5] = {1,2,3,4,5};
PUNT = &X[0]; // APUNTA A X[0].
*PUNT = 9; // [0] = 9
PUNT[3] = 7; // [3] = 7
FOR (I=0; I<5; I++){
 PRINTF("%d,", X[I])); // 9,2,3,7,5 EN PANTALLA
```

```
INT * PUNT, I;

INT X[5] = { 1,2,3,4,5 };
PUNT = X; // X[0]
*X = 11; // X[0] = 11
*(PUNT+3) = 9; // X[0+3] = 9 => X[3] = 9
FOR (I=0; I<5; I++){
PRINTF("%d,", X[I]); // 11,2,3,9,5
```

```
INT *PUNT, I;
INT X[5] = {1,2,3,4,5};
PUNT = X; // X[0]
*(PUNT+2) = 9; // X[0+2] = 9 => X[2] => 9
*(X+3) = 7; // X[3] = 7
PUNT[1] = 11; // X[1] = 11
FOR(I=0; I<5; I++){
 PRINTF("%d,", *(PUNT+I)); // 1, 11, 9, 7, 5
```

EJERCICIO TIPICO DE PARCIAL:

```
STRUCT AUTO{
  INT MODELO;
  INT MARCA[20];
 INT CANT CHOQUES;
} AUTO; // AUTO [20]

VOID MOSTRAR AUTO (STRUCT AUTO *P);

MAIN () { // CARGAR UN AUTO          → STRUCT AUTO * PUNTERO = NULL;
                                          PUNTERO = &AUTO;
AUTO. MODELO = 2000;
AUTO. MARCA = "FIAT"; // STRCPY (AUTO. MARCA)
AUTO. CANTCHOQUES = 4;
// LLAMAR MOSTRAR AUTO
MOSTRAR AUTO (PUNTERO);
}
VOID MOSTRAR AUTO (STRUCT AUTO *P) {
 PRINTF ("%d:", (*P). MODELO); // *P -> MODELO
 PRINTF("%d:", (*P). CANTCHOQUES);
```