UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL RESISTENCIA

EXTENSIÓN ÁULICA GOYA

TRABAJO PRACTICO

“Ejercicios de Repaso sobre structs con vistas al 2do. Parcial”

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN

DOCENTE:

TSP *WIRZ, JORGE.*

-AÑO 2023 -

1- Ejercicios con Structs.

La cláusula typedef como herramienta para simplificar código con structs (entre otros usos.)

¿Cómo funciona typedef en C?

typedef es una palabra reservada en el lenguaje de programación C y C++. Su función es asignar un nombre alternativo a tipos existentes, a menudo cuando su declaración normal es aparatosa, potencialmente confusa o probablemente variable de una implementación a otra. Este operador se usa con frecuencia para abreviar (creando un "alias") los nombres de las estructuras de datos, y para declarar/referenciar los structs definidos previos desde cualquier región del código, evitando tener que hacerlo obligatoriamente en la misma definición del struct.

Definición de tipos: typedef

Se puede dar un nombre nuevo a cualquier tipo de datos mediante typedef.
La sintaxis es

typedef declaración;

Donde declaración tiene la forma de una declaración de variable, sólo que se está definiendo un tipo de datos.

typedef long pareja [2];

define un tipo pareja que se puede usar en declaraciones de variables:

pareja p;

es equivalente a

long p [2];

Ejemplos de código de typedef con estructuras.

```
typedef struct Persona PERSONA;
PERSONA dato;     /* es equivalente a: struct Persona dato; */
```

Un uso típico es la redefinición de tipos estructurados:

```
typedef struct          /* estructura "anónima" */
```

```
{
  char nombre [80];
  char sexo;
  int edad;
} Persona;            /* se declara el tipo Persona */

...

Persona p;  /*acá se aprecia el "alias Persona" para de aquí en adelante definir
              estructuras de tipo Persona en cualquier lugar del código, no sólo
               a continuación (mandatorio, sino) del cierre de } del struct*/

...

p.edad = 44; /* acceso a los campos, etc.*/
```

A continuación, algunos ejercicios resueltos a modo de ejemplo.

Escenario 1: utilizar la cláusula typedef para crear un alias de struct persona: que contenga los campos edad, matrícula (numérica) y nombre con apellido (vectorizar), definiendo una función para la carga de datos y otra que despliegue los datos.

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 3
4 int respuesta, ii;
5
6 struct persona {
7  int edad;
8  char nombre[20];
9  int matricula;
0  };
11 /*Se utiliza la clausura typedef para definir Persona. Note la otra forma de uso*/
12 typedef struct persona Persona;
13 /*Prototipo de funciones para cargar y desplegar datos*/
14 void cargar(int max, Persona *punt); /*punt convierte en puntero a la var punt
15 void desplegar(int max, Persona *punt);
16
17 int main ()
18 {
19 /*Definición del vector alumnos de tipo Persona*/
20  Persona alumnos[MAX];
21 /*Llamada a funciones*/
22  cargar(MAX,alumnos);
```

```
23  desplegar (MAX,alumnos);
24
25  printf("\n\n\n");
26  system("pause");
27  return (0);
28 }
29 /*Primera función*/
30 void cargar(int max, Persona *punt){
31  int ii;
32 /*Ciclo para cargar datos*/
33  for (ii=0; ii<max ; ii++){
34  printf("Ingrese los datos para la %da persona.\n ", ii+1);
35  printf("Ingrese la Edad: \t");
36  scanf("%d", &(*(punt + ii)).edad);
37  printf("\n Ingrese Nombre: \t");
38  fflush(stdin);
39  scanf("%s", &punt[ii].nombre);
40  printf("\n Ingrese la Matricula: \t");
41  scanf("%d", &punt[ii].matricula);
42  }/*Fin del ciclo*/
43 }
44 void desplegar(int max, Persona *punt ){
45  int ii;
46 /*Ciclo para imprimir*/
47  for (ii=0; ii<max ; ii++){
48  printf("\n Datos de la %da persona: ", ii+1);
49  printf("Edad %3d ", (*(punt + ii)).edad);
50  printf(" Nombre %10s ", (*(punt + ii)).nombre);
51  printf(" Matricula %5d\n", (*(punt + ii)).matricula);
52  }/*Fin del ciclo*/
53 }
```

Escenario 2: Definir una estructura que contenga nombre, cedula (o dni) y fecha de nacimiento y un vector con dicho tipo de datos. Definir una función a través de la cual se agregue un nuevo elemento en forma ascendente por fecha de nacimiento.

Definir un menú con la estructura de control do-while que contenga las opciones:

a. Carga "atómica", controlar fin de vector.

b. Despliegue de datos cargados en el vector (con función).

c. Salir del programa.

```
#include <stdio.h>
#include <stdlib.h>
#define TAM 10

struct persona{
char nombre[50];
char cedula[10];
char fecha_nacimiento[15];
 };

typedef struct persona Persona;

int agregar(Persona v1[],int tam,int i);
void desplegar(int i,Persona v1[]);


int agregar(Persona v1[],int tam,int i){
int opcion;
do{
printf("Ingrese nombre de la persona: ");
scanf("%s",v1[i].nombre);
printf("\nIngrese numero de cedula: ");
scanf("%s",v1[i].cedula);
printf("\nIngrese fecha de nacimiento: ");
scanf("%s",v1[i].fecha_nacimiento);

printf("\n¿Quiere seguir cargando datos? \n1.SI\n0.NO\n");
scanf("%d",&opcion);
i++;
 }while(opcion != 0 && i<tam);
 return i;
 }


void desplegar(int i,Persona v1[]){
int j;
for (j=0 ; j<i ; j++){
printf("\t\tPersona Nro %d\n",j+1);
printf("\nNombre: %s",v1[j].nombre);
printf("\nCedula: %s",v1[j].cedula);
printf("\nFecha de Nacimiento: %s\n",v1[j].fecha_nacimiento);
 }
}
```

```
int main()
{
Persona v1[TAM];
int i,j,opcion;
i = 0;
do{
printf("\nMenú de opciones.");
printf("\n1. Cargar datos. ");
printf("\n2. Desplegar datos. ");
printf("\n3. Salir. \n");
scanf("%d",&opcion);

switch(opcion){
case 1 :
i = agregar(v1,TAM,i);
break;
case 2:
desplegar(i,v1);
break;
case 3:
system("pause");
break;
 }
} while(opcion != 3);

return 0;
}
```

Ejercicios para desarrollar.

1 – Modifique el código del Escenario 1 de manera que permita almacenar además una dirección física, una dirección de email, y un nro. de celular.

2 - Modifique el código del Escenario 2, de manera que permita almacenar 20 personas, y además busque y muestre los datos de una persona en base a su cédula (o dni).