

Temas:

- El tipo struct

Objetivos:

Que el estudiante logre...

- Emplear el tipo struct para la definición de registros de datos mediante el análisis y resolución de problemas.

Condiciones de presentación.

- Este trabajo práctico debe realizarse en forma individual.
- El código de los programas solicitados debe estar correctamente rotulado para su identificación, incluyendo comentarios de seguimiento y deberá ser enviado mediante la plataforma CUV.FCEYT hasta el día indicado como límite de presentación.
- La resolución completa de este trabajo práctico, incluyendo diagramas de flujo, deberá pasar a integrar la carpeta de práctica y autoevaluación del alumno.
- **Criterios de evaluación y aprobación.** Este trabajo práctico recibirá una calificación de aprobado o desaprobado. Para aprobar la presentación debe cumplir como mínimo con los siguientes ítems:
- El trabajo práctico debe estar desarrollado completamente.
- La codificación en lenguaje C debe realizarse siguiendo las recomendaciones de la cátedra.
- La presentación del código de los enunciados solicitados deberá realizarse en tiempo y forma mediante la plataforma CUV.FCEYT: <http://cuv.unse.edu.ar>.

Recursos Bibliográficos

- Introducción al Lenguaje C - Santos Espino.
- Introducción a la Programación con C – Marzal.
- Diapositivas, lecciones, disponibles en plataforma del CUV.

Tareas a desarrollar para cada enunciado

- ❖ Desarrolle la diagramación de flujo y la codificación en Lenguaje C de los siguientes enunciados.
- ❖ La resolución de todos los enunciados deberá estar desarrollada en la carpeta de práctica de la asignatura.
- ❖ La entrega de los programas en lenguaje C de los enunciados indicados, deberán ser enviados mediante la plataforma CUV según las fechas y horarios de presentación de cada uno.

TRABAJO PRÁCTICO 3

Enunciado 1

Dado el siguiente código...

Se pide:

- Modifique el código de modo que se ingrese por teclado todos los datos de 5 alumnos y al finalizar se muestre la información de cada uno y el promedio obtenido con sus 4 notas. Utilice el tipo struct.

Enunciado 1 Presentación: código en C.

Fecha: 17/09.

```
#include <stdio.h>

struct alumno {
    char no[25];
    char ap[10];
    int edad;
    int anio;
    int nota[4];
    float prom;
};

int main () {
    int i, acum = 0;
    struct alumno R1 = {"Victor", "Perez", 20, 2020};

    for (i = 0; i < 4; i++) {
        printf ("Ingrese la nota %d:", i + 1);
        scanf ("%d", &R1.nota[i]);
        acum = acum + R1.nota[i];
    }
    R1.prom = (float) acum / 4;
    printf ("El promedio del alumno %s, %s es %2.2f ", R1.ap,
    return (0);
}
```

Enunciado 2

Teniendo en cuenta el siguiente código que se revisó en la lección Arreglos y estructuras, desarrolle un programa modificando dicho código. Utilice otra estructura. Defina esa estructura según su preferencia. Por ejemplo, producto, expediente, legajo, fecha, etc.

Enunciado 2 Presentación: código en C.

Fecha: 17/09.

```
#include <stdio.h>
#define max 2

struct persona {
    char nombre[15];
    char apellido[15];
    int edad;
};

int main () {
    int i = 0;
    struct persona amigo[max];
    for (i = 0; i < max; i++) {
        printf("\nEscriba el Nombre %d:", i + 1);
        scanf("%s", amigo[i].nombre);
        fflush(stdin);
        printf("\nEscriba el Apellido %d:", i + 1);
        scanf("\n%s", amigo[i].apellido);
        fflush(stdin);
        printf("\nEscriba la Edad de %d:", i + 1);
        scanf("\n%d", &amigo[i].edad);
        fflush(stdin);
    }
    printf("\n\nLos registros de Amigos que se introdujeron son: \n\n");
    for(i = 0; i < max; i++){
        printf("%s", amigo[i].nombre);
        printf("\t%s", amigo[i].apellido);
        printf("\t%d", amigo[i].edad);
    }
    return 0;
}
```

Enunciado 3

En una tienda donde solo hay 10 productos se desea calcular el total de ganancia que produjo dicho negocio este mes. Para ello se cuenta por producto con: Precio del Costo, precio de venta, código y cantidad vendida en ese periodo. Realizar un programa que permita calcular las ganancias de la Tienda.

Enunciado 4

Escriba un programa que, tratando una fracción como una estructura de 2 componentes, permita realizar las operaciones de multiplicación y división entre dos fracciones. El programa deberá preguntar la operación a realizar y después los datos para ejecutarla, y repetir estos pasos hasta que se seleccione entre las operaciones la opción terminar.

Enunciado 5

El cálculo de la raíz cuadrada en lenguaje C puede realizarse usando la función sqrt de la librería math.h.

La función sqrt recibe un double representando el número al que queremos sacarle la raíz cuadrada y regresa un número double que indica la raíz cuadrada del número que recibió.

Por ejemplo:

```
#include <math.h>
#include <stdio.h>
int main(int argc, char const *argv[])
{
    double numero = 2525.68;
    double raizCuadrada = sqrt(numero);
    printf("La raíz cuadrada de %lf es %lf\n", numero, raizCuadrada);
    return 0;
}
```

Para elevar un número a una determinada potencia, podemos utilizar la función pow, de la librería math.h. Esta función recibe dos argumentos. El primero es el número al que se desea elevar, y el segundo argumento es la potencia a la que se elevará el número.

Por ejemplo:

```
#include <math.h>
#include <stdio.h>
int main(int argc, char const *argv[])
{
    double numero, potencia; // Aquí guardaremos lo que ingrese el usuario
    printf("Ingresa el numero:\n");
    scanf("%lf", &numero);
    printf("Escribe la potencia para elevarlo:\n");
    scanf("%lf", &potencia);
    double elevado = pow(numero, potencia);
    printf("%lf elevado a la potencia %lf es %lf\n", numero, potencia, elevado);
    return 0;
}
```

Por otra parte, para calcular la distancia entre dos puntos en el plano la fórmula matemática es la siguiente:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Un punto en el plano se puede representar mediante una estructura con dos campos. Un campo para el eje de las x y un campo para el eje de las y.

Se pide:

Desarrollar un programa que dados dos puntos calcule la distancia entre ellos. La tarea se repite hasta que el usuario indique terminar.

Utilice el tipo struct para representar a los puntos.

Enunciado 5 Presentación: diagrama de flujo y codificación en C.

Fecha: 17/09.