

Temas: Programación avanzada en Lenguaje C: Punteros. Tipos estructurados (struct). Funciones.

Objetivos: que el alumno demuestre...

- Conocer las características de las funciones de biblioteca para la Entrada - Salida, para manejo de caracteres y strings, y salida o entrada formateada.
- Habilidad para aplicar adecuadamente el uso y manipulación de datos, así como estructuras de datos básicas tales como arreglos usando lenguaje C.
- Capacidad y destreza en la aplicación de variables punteros y registros de datos mediante struct en la resolución de problemas.
- Reconocer la importancia de las funciones en lenguaje C y la diferencia entre prototipos de función, definición de funciones y invocación de funciones.
- Destreza en la programación en lenguaje C de prototipos de función, definición de funciones y invocación de funciones.

Modalidad:

El taller se llevará a cabo durante un período aproximado de 10 días, el trabajo realizado podrá ser enviado hasta el día domingo 2 de octubre y defendido en sesión presencial a partir del 3 de octubre. Mediante la herramienta Taller de la plataforma del CUV el alumno recibirá una tarea a desarrollar y, en una segunda etapa, se le asignará aleatoriamente la tarea desarrollada por otro compañero sobre la cual deberá aplicar análisis de los criterios usados en la resolución del problema, verificación de código, y aporte de correcciones y/o comparaciones con su propio trabajo.

Inicio: 18 de septiembre de 2022.

Finalización: 2 al 6 de octubre de 2022.

Instancia recuperatoria: será estipulada de acuerdo al trabajo presentado en primera instancia, y consistirá en su elaboración, corrección y/o adaptación.

Condiciones de presentación.

- Este trabajo debe realizarse en forma individual en su primera etapa y colaborativamente con un compañero en su segunda etapa.
- Los diagramas de flujo deben estar completamente desarrollados en la CARPETA DE LA ASIGNATURA, de manera prolija, e incluyendo las descripciones necesarias para su mejor seguimiento (identificación, variables utilizadas y sus funciones, etc.).
- El código del programa solicitado 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 *Fin de los envíos*.
- La tarea de evaluación sobre el trabajo del compañero deberá seguir los criterios indicados a tal fin, y ser enviada hasta la fecha indicada como *Fin de las evaluaciones*.
- La resolución completa de este trabajo, incluyendo diagramas de flujo y la evaluación realizada sobre el trabajo del compañero, deberá pasar a integrar la carpeta de práctica y autoevaluación del alumno.
- El archivo enviado deberá ser un archivo compactado conteniendo en formato PDF los diagramas de flujo, y en archivo con extensión .C el código desarrollado en lenguaje C. No se aceptarán envíos que no cumplan con este formato.

Criterios de evaluación y aprobación.

Se aplicará una valoración cuantitativa politómica, con escala de valoración del 1 al 100.

El puntaje mínimo para aprobar es de cincuenta (50) puntos, sobre una calificación máxima de cien (100) puntos.

Para aprobar la presentación debe cumplir con los siguientes ítems:

PRIMERA ETAPA

- El trabajo solicitado debe estar desarrollado completamente.
- La codificación en lenguaje C debe realizarse siguiendo las recomendaciones de la cátedra.
- El desarrollo de los diagramas de flujo debe realizarse cumpliendo las indicaciones relativas a la diagramación estructurada y modular.
- La presentación de la diagramación de flujo y el código solicitado en la primera etapa deberá realizarse en tiempo y forma mediante la plataforma CUV.FCEYT: <http://cuv.unse.edu.ar>, desde el día indicado como *Comienzo de los envíos* hasta el día indicado como *Fin de los envíos*.

SEGUNDA ETAPA

- En la segunda etapa el alumno deberá evaluar el trabajo realizado por un compañero. Esta tarea se realizará desde el día indicado como *Comienzo de las evaluaciones* hasta el día indicado como *Fin de las evaluaciones*.

- Para realizar la actividad de esta segunda etapa el alumno deberá tener en cuenta los siguientes **Criterios de Evaluación**:
 - Sobre el código del programa:
 - Organización del código del programa: Observe la organización del código del programa e identifique sus diversos componentes.
Determine el nivel de organización del programa considerando la organización aplicada en cada una de las estructuras de control utilizadas en la resolución del problema.
 - Legibilidad y comprensión del código del programa: Identifique si el código del programa está indentado permitiendo seguir adecuadamente cada una de las estructuras de control de flujo.
Observe si los nombres utilizados en el código permiten un fácil seguimiento de las sentencias.
Observe la utilización de comentarios en el código del programa y determine si cumplen la función de aclarar y señalar detalles que ayuden a la lectura del código.
 - Sobre la estrategia de resolución del problema:
 - Análisis de la eficacia/eficiencia de la estrategia elegida para la resolución del problema: Identifique en qué medida el código del programa cumple con las solicitudes planteadas en el enunciado propuesto para el trabajo.
Determine si la estrategia elegida es la más adecuada, según su criterio.
 - Nivel de creatividad del algoritmo de solución aplicado: ¿La estrategia utilizada para resolver el problema es novedosa? o considera que se trata de una estrategia común y sin elementos que la distingan entre otras alternativas de solución?
Determine el nivel de creatividad que identifica en la estrategia de solución aplicada.
 - Sobre el uso del lenguaje C y las bibliotecas de funciones:
 - Adecuada selección de variables y sus tipos de datos.
 - Correcto uso de variables en expresiones y en conversiones de tipo.
 - Correcto uso de funciones de entrada salida formateada, y de manejo de caracteres y strings.
 - Uso adecuado y eficiente de la estructura de datos arreglo.
 - Correcta utilización de sentencias condicionales y ciclos en la aplicación de la programación estructurada.
 - Comparación con su propio trabajo:
 - Comparaciones con su propio trabajo: Tenga en cuenta la estrategia de solución que aplicó en su propio trabajo y determine en qué medida este trabajo evaluado es más/menos eficiente, más/menos eficaz.
 - Correcciones y/o recomendaciones de mejoramiento: ¿Identifica correcciones para realizar?
¿Recomendaría puntos o aspectos a mejorar en la resolución elegida en el trabajo evaluado?
 - Sobre el grado de resolución del problema propuesto:
 - Grado de resolución del problema propuesto: Determine el grado o nivel en que el problema propuesto fue correctamente resuelto mediante la estrategia de solución planteada en el trabajo evaluado.

Fechas de presentación: Envíos al CUV según las fechas indicadas a continuación:

PRIMERA ETAPA

- Comienzo de los envíos: jueves 21 de septiembre, 0:00 horas.
- Fin de los envíos: viernes 29 de septiembre, 23:55 horas.

SEGUNDA ETAPA

- Comienzo de las evaluaciones: sábado 30 de septiembre, 0:00 horas.
- Fin de las evaluaciones: domingo 01 de octubre, 23:55 horas.

Recursos Bibliográficos

Lenguaje C. KERNIGHAN & RITCHIE. Prentice – Hall. Hispanoamericana S.A.
Como programar en C/C++ 2º Edición. DEITEL HARVEY. Prentice – Hall. Hispanoamericana S.A.
Introducción al Lenguaje C, de ESPINO, archivo pdf
Introducción a la Programación en C, ANDRES MARZAL - ISABEL GRACIA. Departamento de Lenguajes y Sistemas Informaticos Universitat Jaume I, archivo pdf

Resuelva el siguiente enunciado usando **diagramación estructurada** y codificando en Lenguaje C.

Debe aplicar Punteros. Tipos estructurados (struct). Funciones.

❖ Enunciado

Realizar un programa que pida al usuario la información de 10 empleados de una empresa.

Para cada empleado se almacenará la información de su legajo y su nombre. El legajo del empleado es un número entero en el rango de 1000 a 9999.

El programa deberá identificar aquellos empleados cuyos legajos sean múltiplos de un número ingresado por teclado.

Según la orden del usuario, se mostrará la información de todos los empleados, sólo de aquellos que cumplen la condición de legajos múltiplos, o sólo aquellos que no cumplen con la condición. Deberá usarse un formato de columna.

Para llevar a cabo la organización de las actividades, deberá utilizarse un menú como el siguiente:

Taller 1: Empleados

- 1- Ingresar empleados
- 2- Mostrar todos
- 3- Mostrar los legajos múltiplos
- 4- Mostrar los legajos no múltiplos

Ingrese su opción:

En la resolución de este ejercicio se deben usar, como mínimo, las siguientes funciones:

a) *void leerEmpleado(int *legajo, char *nombre);*

Objetivo: Solicita los datos relativos a un empleado y verifica que sean válidos.

b) *int controlarLegajo(int legajo, int valorIngresado);*

Valor de retorno: 1 si el legajo es múltiplo de valorIngresado y 0 en caso contrario.

c) *void mostrarEmpleado(struct Empleado emp);*

Objetivo: muestra los datos de un solo empleado. Se utiliza para generar el listado encolumnado.

❖ Ejemplo

El usuario ingresa el empleado Roberto, con legajo 21300. El legajo es rechazado debido a que no cumple con el rango.

El usuario ingresa el empleado Roberto, con legajo 2130. Los datos son controlados y aceptados.

Se ingresa el empleado nombre en blanco, legajo igual a 500. Los datos son rechazados.

Se ingresa el empleado Juan, con legajo 6233. Los datos son controlados y aceptados.

Se solicita imprimir aquellos empleados cuyos legajos sean múltiplos de 10, la salida será:

TITULO (indicar título)

LEGAJO	NOMBRE
Roberto	2130
José	5520
...etc.	

TALLER 1

ANEXO

Una rúbrica es una guía precisa que valora los aprendizajes y productos realizados. Es una tabla que desglosa los niveles de cumplimiento de una tarea, con criterios específicos.

A continuación se presenta la rúbrica que será utilizada, tanto por estudiantes como por los docentes, para evaluar este Taller.

RÚBRICA PARA TALLER 1			
CRITERIO 1 Sobre el código del programa: <ul style="list-style-type: none">Organización del código del programa: Observe la organización del código del programa e identifique sus diversos componentes. Determine el nivel de organización del programa considerando la organización aplicada en cada una de las estructuras de control utilizadas en la resolución del problema.Legibilidad y comprensión del código del programa: Identifique si el código del programa está indentado permitiendo seguir adecuadamente cada una de las estructuras de control de flujo. Observe si los nombres utilizados en el código permiten un fácil seguimiento de las sentencias. Observe la utilización de comentarios en el código del programa y determine si cumplen la función de aclarar y señalar detalles que ayuden a la lectura del código.			
(5) El código se encuentra muy poco organizado, sin indentación adecuada, sin variables usadas cuyos nombres sean nemotécnicos y casi no se utilizaron comentarios para aclarar y explicar los módulos del programa.	(10) El código se encuentra organizado en algunas áreas del programa, muchas líneas no fueron indentadas, cuenta con variables usadas cuyos nombres no son nemotécnicos y se utilizaron pocos comentarios para aclarar y explicar los módulos del programa.	(15) El código se encuentra generalmente organizado, la mayoría de las líneas fueron indentadas, la mayoría de las variables usadas son nemotécnicas y se utilizaron suficientes comentarios para aclarar y explicar todos los módulos del programa.	(20) El código se encuentra correctamente organizado, indentando, todas las variables usadas son nemotécnicas y se utilizaron comentarios para aclarar y explicar todos los módulos del programa.
CRITERIO 2 Sobre la estrategia de resolución del problema: <ul style="list-style-type: none">Análisis de la eficacia/eficiencia de la estrategia elegida para la resolución del problema: Identifique en qué medida el código del programa cumple con las solicitudes planteadas en el enunciado propuesto para el trabajo.Determine si la estrategia elegida es la más adecuada, según su criterio.			
(2) Se cumple con menos del 30% de las solicitudes indicadas en el enunciado del problema y la estrategia elegida no es la adecuada.	(4) Se cumple con menos del 50% de las solicitudes indicadas en el enunciado del problema y la estrategia elegida no es tan adecuada.	(7) Se cumple casi totalmente con las solicitudes indicadas en el enunciado del problema y la estrategia elegida es suficiente.	(10) Se cumple totalmente con las solicitudes indicadas en el enunciado del problema y la estrategia elegida es adecuada.

TALLER 1

RÚBRICA PARA TALLER 1			
CRITERIO 3 Sobre el uso del lenguaje C y las bibliotecas de funciones: <ul style="list-style-type: none"> Adecuada selección de variables y sus tipos de datos. Correcto uso de variables en expresiones y en conversiones de tipo. Correcto uso de funciones de entrada salida formateada, y de manejo de caracteres y strings. Uso adecuado y eficiente de la estructura de datos arreglo. Correcta utilización de sentencias condicionales y ciclos en la aplicación de la programación estructurada. 			
(5) No se seleccionaron adecuadamente variables y tipos de datos, no se usaron de manera correcta variables definidas y funciones, no se usó adecuadamente la estructura de datos arreglo y se usaron a veces bien las sentencias condicionales así como los ciclos en la programación estructurada.	(15) Se seleccionaron adecuadamente muy pocas variables y tipos de datos, se usaron de manera correcta pocas variables definidas y funciones, muy pocas veces se usó adecuadamente la estructura de datos arreglo y se usaron a veces bien las sentencias condicionales así como los ciclos en la programación estructurada.	(30) Se seleccionaron adecuadamente algunas las variables y tipos de datos, se usaron de manera correcta algunas variables definidas y funciones, generalmente se usó adecuadamente la estructura de datos arreglo y se usaron generalmente bien las sentencias condicionales así como los ciclos en la programación estructurada.	(40) Se seleccionaron adecuadamente las variables y los tipos de datos, se usaron de manera correcta las variables definidas y las funciones, se usó adecuadamente la estructura de datos arreglo y se usaron de forma correcta las sentencias condicionales así como los ciclos en la programación estructurada.
CRITERIO 4 Comparación con su propio trabajo: <ul style="list-style-type: none"> Comparaciones con su propio trabajo: Tenga en cuenta la estrategia de solución que aplicó en su propio trabajo y determine en qué medida este trabajo evaluado es más/menos eficiente, más/menos eficaz. Correcciones y/o recomendaciones de mejoramiento: ¿Identifica correcciones para realizar? ¿Recomendaría puntos o aspectos a mejorar en la resolución elegida en el trabajo evaluado? 			
(2) Este trabajo es muy malo comparado con el mío.	(4) Este trabajo no es mejor que el que he desarrollado	(7) Este trabajo es tan bueno como el que he desarrollado.	(10) Este trabajo es más eficiente que el que desarrollé.
CRITERIO 5 Sobre el grado de resolución del problema propuesto: <ul style="list-style-type: none"> Grado de resolución del problema propuesto: Determine el grado o nivel en que el problema propuesto fue correctamente resuelto mediante la estrategia de solución planteada en el trabajo evaluado. 			
(2) Este trabajo fue correctamente resuelto en un 10%.	(5) Este trabajo fue correctamente resuelto en un 30%.	(10) Este trabajo fue correctamente resuelto en un 80%.	(20) Este trabajo fue correctamente resuelto en un 100%.

