

Proyecto: Diseño, codificación y presentación en clase

Actividad por equipo:

Diseño en pseudocódigo de Pselnt o diagrama de flujo de DFD o Pselnt, codificación en lenguaje C cumpliendo con las reglas y equivalencias acordadas en clase y presentación en el salón.

Reglas:

La actividad se realizará estrictamente por equipos respetando la organización asignada.

Dos equipos escogerán un algoritmo de la lista mostrada. (No se pueden repetir)

Una vez seleccionado el algoritmo, un equipo realizará el diseño en pseudocódigo o diagrama de flujo con la herramientas Pselnt o DFD y el otro equipo aplicará dicho diseño a codificación en lenguaje C.

La codificación en C deberá cumplir por completo con el diseño del otro equipo a excepción de funciones que sean propias del lenguaje C que serán simplemente indicadas como comentario o impresión en el diseño.

1. Una tienda de videojuegos divide sus mostradores en categorías por precios de los más baratos a los más caros, la primera categoría corresponde a videojuegos de diferentes consolas con un precio de 199 pesos, la segunda con un costo de 299 pesos, la tercera de 399 pesos, la cuarta de 499 pesos y la quinta con estrenos en 1099 pesos. Diseñe y codifique un algoritmo que solicite el nombre del cliente y cuántos videojuegos desea comprar de cada categoría imprimiendo tanto en consola como en un archivo de texto plano el nombre del cliente, la suma de cada categoría y el total a pagar. Considerar las verificaciones necesarias como cantidades positivas y no excesivas para categoría para evitar venta al mayoreo o no disponible en inventario.
2. Una tienda de deportes tiene una oferta de fin de semana en la que en la compra de tres artículos el de menor valor tiene el 50% de descuento y el que le siga en menor precio tiene el 25%. El algoritmo a diseñar y codificar debe pedir nombre al usuario, el valor de los artículos, calcular los descuentos correspondientes e imprimir al final tanto en consola como en un archivo de texto plano el nombre del cliente, el precio de cada artículo después del descuento y el monto total. Considerar las verificaciones necesarias como cantidades positivas y que pueden ser iguales o diferentes.
3. Cómo atrapar un pokemon desde el punto de vista del jugador tomando en cuenta los puntos de combate del pokemon, rareza, disponibilidad de tipos de pokebolas y bayas. Simular si fue atrapado o no o si escapó empleando números aleatorios. Imprimir en consola desde una función alguna figura de pokemon mediante letras, números o caracteres especiales.
4. Ruleta sangrienta con un arreglo de cadenas para los nombres de los participantes que serán ingresados por el usuario, emplear números aleatorios para seleccionar al afortunado y tiempo para retardar el resultado. Cargar los nombres de un archivo de texto plano.
5. A partir de 2 arreglos bidimensionales leer los datos necesarios de un equipo de fútbol y calcular el promedio de las estadísticas de cada jugador. Un arreglo será de caracteres con dimensiones dadas en renglones de acuerdo a los integrantes del equipo y columnas por la longitud máxima posible de sus nombres. El otro arreglo será numérico con igual cantidad de renglones que el anterior más uno (donde se guardarán los promedios de cada columna) y el número de columnas será el necesario para almacenar las estadísticas de cada integrante como jugadores de fútbol relacionadas a partidos

jugados, goles, asistencias, tarjetas amarillas, tarjetas rojas y faltas. Los nombres de los jugadores serán cargados de un archivo de texto plano y de otro archivo las estadísticas de cada uno.

'L'	'u'	'l'	'u'	'\0'		
'J'	'a'	'v'	'i'	'\0'		
'J'	'o'	'r'	'g'	'e'	'\0'	
'R'	'a'	'f'	'a'	'\0'		

Promedios



<i>PJ</i>	<i>G</i>	<i>Asis</i>	<i>TA</i>	<i>TR</i>	<i>F</i>

6. Instalación de Linux junto a Windows (1 equipo hace video de descarga de iso y quema en cd o usb y el otro equipo la instalación). Emplear la distribución Ubuntu para escritorio en sus versiones 14.04 o 16.04 de acuerdo a la potencia del equipo de prueba. Mostrar algunas actividades de forma rápida como la instalación de programas desde el centro de Software de Ubuntu, cómo emplear LibreOffice, cómo comprimir archivos o directorios, visualizar imágenes y archivos pdf, reproducir contenido multimedia de audio y video.

Entrega:

Impreso y PDF con portada, captura de pantalla del diseño en pseudocódigo o diagrama de flujo, prueba de escritorio así como su pantalla de ejecución, captura de pantalla del código en lenguaje C desde la ventana del IDE o editor de texto plano empleado (Dev-C++, CodeBlocks, Notepad++, gedit) con cada línea comentada y su respectiva pantalla de ejecución. Recuerden agregar al código en C la impresión en pantalla del recuadro al inicio con la descripción del programa y el nombre de quien lo hizo, líneas para resaltar resultados y otros detalles de estilo con código ASCII como los mostrados en la presentación. (Excepto el ejercicio 6)

Se tomarán video presentando los integrantes de cada equipo y la actividad que realizaron, las ejecuciones del diagrama o pseudocódigo así como del programa completo y lo subirán a youtube o alguna otra plataforma. Juntar los videos de los dos equipos para el mismo ejercicio en uno que no exceda los 5 minutos de duración.

El ejercicio se calificará por todos en el salón posterior a la reproducción de su video y exposición por parte de dos integrantes de cada equipo de forma breve (10 minutos) acerca del proceso de desarrollo y su experiencia.

Enviarán por correo electrónico el pdf, los archivos .c, de proyecto .dev, de PseInt con los diagramas de flujo o pseudocódigo y un archivo README.txt con los datos de los integrantes del equipo y la liga del video. Colocar en una carpeta todos los archivos mencionados y comprimir en .zip dando por nombre el número de ejercicio asignado y el grupo (p. ej. Ejercicio1_gpo9.zip) con ese mismo asunto.

labs.huicho@gmail.com
asunto: Ejercicio1_grupo#