

Maestría en Tecnología de Cómputo

Curso propedéutico de Programación

Tema III: Estructuras de datos.

Clase Práctica No. 3

Título: Ejercicios sobre manejo de estructuras de datos estáticas. Recursividad, búsqueda y ordenamiento. Solución de problemas sencillos.

Ejercicios a Resolver en la Clase

Ejercicio 1.

Escriba un programa que invierta, o sea que ponga en orden inverso los elementos de un arreglo dado y a continuación imprima el arreglo ya invertido en pantalla. No utilice un arreglo auxiliar.

Ejercicio 2.

En una empresa comercializadora de equipos electrodomésticos se lleva el control del gasto diario por concepto de compras de productos a suministrar y de los ingresos diarios por las ventas de productos a los diferentes clientes. Se conoce el saldo de la empresa después de cerrar la última operación del mes anterior (mes = 30 días). Se desea conocer el saldo que queda al finalizar el mes actual. Escriba un programa que permita que el usuario entre por teclado el saldo del mes anterior y los datos de los gastos e ingresos a dos arreglos de 30 elementos cada uno. Luego debe imprimir el saldo al final del mes actual.

Ejercicio 3.

Realice un programa que dado una matriz y sus dimensiones, devuelva su traspuesta.

Ejercicio 4. Elecciones.

Realice un programa que permita a cada consejo electoral llevar una serie de estadísticas sobre el estado de las elecciones. El programa debe recibir la cantidad de electores y luego la votación de cada elector. El formato del voto es un número del 1 al 4, que representa el número del partido a seleccionar. Cada vez que se realice una votación se debe mostrar el porcentaje de los votos de cada partido, el porcentaje de los electores que ya han votado y cuántos faltan por votar. Cuando voten todos los posibles electores debe mostrarse un mensaje que diga: "Han terminado los comicios".

Ejemplo:

Número de electores: 2

Votar aquí: 1

1 - 100 % de los votos
2 - 0 %de los votos
3 - 0 %de los votos
4 - 0 %de los votos
Ha votado el 50 % de los electores
Faltan por votar 1 electores
Votar aquí: 5
El formato del voto es 1, 2, 3 o 4
Votar aquí: 3
1 - 50 %de los votos
2 - 0 %de los votos
3 - 50 %de los votos
4 - 0 %de los votos
Ha votado el 100 % de los electores
Faltan por votar 0 electores
Han terminado los comicios

Ejercicio 5. Cabaret Piña Colada.

En el Cabaret Piña Colada se dan diariamente un máximo de veinte reservaciones para ocupar 5 mesas de 4 personas. La capacidad máxima para una reservación es de 4 personas (una mesa). Obviamente los miembros de una reservación no querrán sentarse en mesas separadas, de manera que si se realiza una reservación y hay una mesa parcialmente ocupada, se completa si la cantidad de miembros de la reservación es menor o igual a la capacidad disponible en la mesa. Puede ocurrir que en todas las mesas haya capacidades y no se pueda realizar una determinada reservación.

Realice un programa que permita introducir una reservación (cantidad de personas) y devuelva la mesa asignada. Si no hay mesa disponible para la reservación, se debe mostrar un mensaje que diga "No hay mesa disponible para la reservación". Mientras haya capacidades en el cabaret habrá posibilidad de reservar hasta que se acaben, en ese caso se debe mostrar un mensaje que diga: "Ya no quedan capacidades".

Ejemplo:

Reservación para (x) personas: 1
Usted ha reservado en la mesa 1
Capacidad disponible en el cabaret: 19
Reservación para (x) personas:4
Usted ha reservado en la mesa 2
Capacidad disponible en el cabaret: 15
Reservación para (x) personas:1
Usted ha reservado en la mesa 1
Capacidad disponible en el cabaret: 14
Reservación para (x) personas:5
La Reservación es limitada para 4 personas
Reservación para (x) personas:4
Usted ha reservado en la mesa 3
Capacidad disponible en el cabaret: 10

Reservación para (x) personas:4
Usted ha reservado en la mesa 4
Capacidad disponible en el cabaret: 6
Reservación para (x) personas:4
Usted ha reservado en la mesa 5
Capacidad disponible en el cabaret: 2
Reservación para (x) personas:4
No hay mesa disponible para la reservación
Capacidad disponible en el cabaret: 2
Reservación para (x) personas:2
Usted ha reservado en la mesa 1
Capacidad disponible en el cabaret: 0
Ya no quedan capacidades

Ejercicio 6. Torre de Hanói.

Todos los científicos de computo incipiente deben enfrentarse con ciertos problemas clásicos, y las torres de Hanói, es uno de los más famosos. Dice la leyenda que un templo del Lejano Este, los monjes están intentando mover una pila de discos de una estaca hacia otra. La pila inicial tenía 64 discos ensartados en una estaca y acomodados de la parte superior a la parte inferior en tamaño creciente. Los monjes están intentando mover la pila de esta estaca a la segunda con las limitaciones que solo puede ser movido un disco a la vez, y en ningún momento se puede colocar un disco de mayor tamaño sobre uno de menor. Existe una tercera estaca disponible para almacenamiento temporal de discos.

Supongamos que los monjes están intentando mover los discos de la estaca 1 a la 3. Se desea un algoritmo que imprima la secuencia precisa de las transferencias disco a disco entre estacas.

Si fuéramos a enfocar este problema con métodos convencionales, nos encontraríamos rápidamente enmarañados y sin esperanza de poder manejar los discos. En vez de ello si atacamos el problema de manera recursiva, de inmediato se vuelve manejable. El mover n discos puede ser visualizado en términos de mover $n-1$ discos (y de ahí la recursión) como sigue:

- 1- Mover $n-1$ discos a la estaca 2, utilizando la estaca 3 como almacenamiento temporal.
- 2- Mover el último disco (el más grande de la estaca 1 a la estaca 3)
- 3- Mover los $n-1$ discos de la estaca 2 a la 3, utilizando la estaca 1 como almacenamiento temporal.

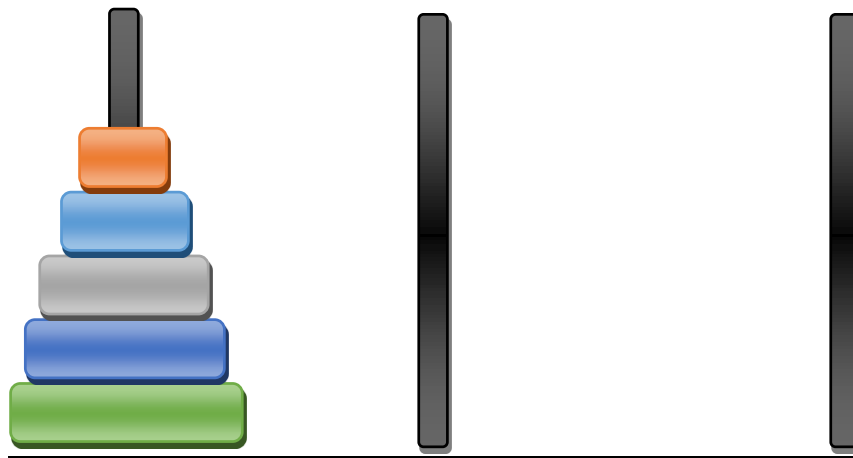
El proceso termina cuando la última tarea consiste en mover el disco $n=1$, es decir, el caso base. Esto se lleva a cabo en forma trivial, moviendo el disco, sin necesidad de utilizar el área temporal de almacenamiento. Escriba un programa para resolver el problema de las Torres de Hanoi. Utilice una función recursiva con cuatro parámetros:

1. El número de discos a moverse.

2. La estaca en la cual se acumularan estos discos al inicio.
3. La estaca en la cual esta pila de discos se moverá.
4. La estaca a usarse como área temporal.

Su programa deberá imprimir las instrucciones precisas que deberán seguirse para mover los discos de la estaca de arranque a la estaca de destino. Por ejemplo, para mover una pila de tres discos de la estaca 1 a las estaca 3, su programa deberá imprimir la siguiente serie de movimientos:

1-->3(estosignificamoverundisco de la estaca 1 a la estaca 3)
 1-->2
 3-->2
 1-->3
 2-->1
 2-->3
 1-->3.



Ejercicio 7. Coeficientes polinomiales

Realice un programa que dado la potencia polinomial ($n < 13$) del polinomio $\mathbf{p} = (\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 + \dots + \mathbf{x}_k)^n$, la cantidad de variables ($k < 13$) y k enteros no negativos (n_1, n_2, \dots, n_k) donde $n_1 + n_2 + \dots + n_k = n$. Devuelva el coeficiente del monomial $x_1^{n_1} x_2^{n_2} x_3^{n_3} \dots x_k^{n_k}$.

Acercamiento a la Solución

El coeficiente se calculan a partir de :

$$C = \frac{n!}{n_1! n_2! n_3! \dots n_k!}$$