

1) (DEBE RESOLVERSE EN PYTHON)**Navegando por el mundo...**

La empresa Porto Siciliano S.A. nos solicita la realización de un programa que procese las embarcaciones que llegan a su puerto de Taormina, y contabilizar el costo por el uso del puerto compuesto por un costo fijo + CTE x metros de eslora (costo fijo = 2000, CTE= 33,22)

La información será suministrada desde un archivo de texto, llamado nav.txt. Este archivo tendrá el siguiente formato.

Viaje Nro, Empresa Naviera, Origen, Horario de Llegada, cantidad de personas a bordo, eslora, Estado (estimado, demorado, arribando)

nav.txt

Costa7565, Costa Cruceros, Barcelona, 0705, 2500, 250, estimado

MSC8796, MSC Cruceros, Atenas, 0730, 1900, 198, estimado

MSC8700, MSC, Barcelona, 0710, 1500, 245, demorado

IT53, Naviera Italo, Nápoles, 0650, 33, 87, arribando

MS932, Mediterranean Shipping, Suez Egipto, 0655, 15, 200, arribando

CMA43, CMA CGM, Melilla Marruecos, 0735, 33, 244, demorado

....

Para lo cual el programa pedirá:

- a- Procesar el archivo "nav.txt" y volcarlo en una estructura que luego le permita iniciar la resolución del resto de puntos.
- b- Determinar el top 3 de ciudades son las más utilizadas como puertos de origen para cualquiera de los cruceros que llegan a Taormina.
- c- Buscar todos los arribos entre las 0700 y las 0745 y que estén demorados. Desplegar Viaje nro y eslora, ordenado por eslora de mayor a menor.
- d- Permitir al usuario ingresar una empresa naviera y que el programa liste por pantalla toda la información de los viajes que tienen una cantidad de pasajeros mayor al promedio general de todos los viajes.
- e- Crear 3 archivos de textos uno por cada estado, que contenga la información de todas las embarcaciones ordenadas por horario. La última línea de cada archivo debe indicar el costo total de operación portuaria. Los archivos se deberán llamar, estimado.txt, demorado.txt y arribando.txt

Aclaración 1: Los puntos a,b,c,d,e deberán resolverse con funciones.

Aclaración 2: Es obligatorio usar try-except por lo menos en un lugar y justificar su uso

Aclaración 3: Debe existir un menú para poder llamar a las opciones a gusto del usuario

2) (DEBE RESOLVERSE EN C)

Dada la siguiente firma de función:

bool ksum(int[] vector, int n, int k); // Evalúa si existe, o no, dos índices distintos del vector cuyos valores sumados sean igual a k

Se pide completar dicha función y utilizarla en un programa para imprimir por pantalla si existe, o no, índices distintos del vector que cumplan dicha suma.

Ejemplos:

1er ejemplo:

VECTOR = [1,2,3,4,5,7,8,9], K = 5

En este caso pueden ser más de dos i,j que cumplan que $VECTOR[i] + VECTOR[j] = K$.

$VECTOR[0] + VECTOR[3] = 1 + 4 = 5$

$VECTOR[1] + VECTOR[2] = 2 + 3 = 5$

Pero basta con la primera ocurrencia para finalizar con la evaluación.

2do ejemplo:

VECTOR = [1,4,6,2,6,8,5], K = 9

En este caso...

$VECTOR[6] + VECTOR[1] = 5 + 4 = 9$