

La empresa aseguradora SuSeguro necesita elaborar un programa para procesar los registros y elaborar informes estadísticos sobre el trabajo de sus vendedores de seguros en el área automotores.

Los vendedores venden seguros destinados a cubrir las sumas aseguradas (ValorAsegurado).

Este valor se fija en las pólizas extendidas a los asegurados.

Si los asegurados sufren siniestros (robos, incendios, destrucción parcial o total o lo que fuere) sobre los bienes cubiertos por los seguros, la empresa Su Seguro les abona la cifra pactada en el contrato de seguro.

Los datos se guardan en los archivos: “Vendedores.dat”, “Polizas.dat” y “Siniestros.dat”, según los registros que se describen a continuación:

//Nota **palabra** es un typedef correspondiente a un string de 30 caracteres

//stringpat typedef correspondiente a un string de 8 caracteres, usado para la patente

```
struct Vendedor
{
int IDVendedor;
palabra ApellidoVendedor;
float TasaRendimiento ;
};
```

```
struct Poliza
{
int NroPoliza;
int IDVendedor;
palabra Asegurado;
palabra Marca;
stringpat Patente;
float ValorAsegurado;
};
```

```
struct Siniestro
{
int IDSiniestro;
int NroPoliza;
float ValorSiniestrado ;
};
```

Cuando se produce un siniestro, la empresa realiza el siguiente cálculo:

Por cada vendedor se calcula la tasa de rendimiento como el porcentaje indicado de la sumatoria de los siniestros en la sumatoria del capital asegurado, de la siguiente forma:

$$\text{Tasa de Rendimiento} = (\text{Sumatoria ValorAsegurado} / \text{Sumatoria ValorSiniestrado}) * 100 / \text{Total CapitalAsegurado}$$

Esto se usa para premiar con una suba en sus comisiones para el siguiente ejercicio, a los vendedores que obtengan una tasa superior al 5% en el rendimiento.

Deberán realizarse las siguientes funciones:

void VerificarArchivos() - Función que se asegura de la existencia de los archivos antes de comenzar el trabajo diario. Si los archivos no existen, los crea. Emite mensajes indicando lo que sucede con cada archivo.

void CargarDatos(Archivo) – Función que agrega datos al archivo que corresponda, pasado como parámetro.

int ValidarVendedor(char ApellidoProductor[]) – Función que valida la carga de pólizas solo a vendedores verificados como existentes. Busca al vendedor en cuestión y, si lo encuentra en el archivo devuelve su IDVendedor, de lo contrario devuelve - 1 (menos uno).

int ValidarPoliza(int NroPoliza) – Función que valida la carga de siniestros solo a pólizas verificadas como existentes, si encuentra la póliza en el archivo devuelve 1 (uno), de lo contrario devuelve - 1 (menos uno).

float CalcularTasaRendimiento(int IDVendedor) – Función que calcula los porcentajes de rendimiento de las pólizas vendidas por cada productor antes de informar. Debe trabajar con los 3 archivos abiertos a la vez para entrecruzar los datos guardados y efectuar todos los cálculos necesarios. Se puede (y es recomendable) programar funciones auxiliares.

$$\text{Tasa de Rendimiento} = (\text{Sumatoria ValorAsegurado} - \text{Sumatoria Valorsiniestrado}) * 100 / \text{Total CapitalAsegurado}$$

donde:

Tasa de Rendimiento: es el campo a calcular en la estructura productor.

Sumatoria ValorAsegurado: es la sumatoria de todos los capitales asegurados por las pólizas del productor.

Sumatoria ValorSiniestrado: es la sumatoria de todos los pagos de siniestros del productor.

Total, CapitalAsegurado: es la sumatoria de todas las pólizas que emitió la compañía en el ejercicio.

Funciones para presentar informes al operador, en forma de listados por pantalla o impresora:

void ListarInfo(Archivo) : Muestra el contenido del archivo pasado como parámetro (cualquiera de los tres).

void ListarPolizasVendedor(palabra ApellidoVendedor) : Muestra todas las pólizas y todos los siniestros de un vendedor pasado como parámetro.

void VendedoresAPremiar() : Muestra los vendedores que han tenido un porcentaje de rendimiento inferior al 5% en la tasa de rendimiento.

void AseguradoPiedra() - Muestra toda la información disponible del asegurado al que más dinero debió pagársele por siniestros durante el ejercicio anual.

Realizar una función main() que permita ejecutar el programa.

Tener en cuenta:

Un vendedor tiene cero o muchos asegurados. Cada seguro tiene una póliza.

Una póliza puede tener cero o varios siniestros. Un asegurado puede tener uno o varios seguros.

Las pólizas que figuran en nuestros archivos están todas vigentes.

Nota: el trabajo practico es de carácter obligatorio e individual y deberá estar completamente funcional. No se admitirá entregas parciales del trabajo practico.

Esta es una de las condiciones para tener aprobada la cursada. La defensa de este trabajo será en fecha de final