|  |
| --- |
| 2do Parcial Algoritmos y estructura de datos 2020 |
| Apellido y Nombre: Legajo: |
| Debe desarrollar solo uno de los puntos según el valor del digito verificador del legajo  1,2 🡪 Punto 5; 3,4 🡪 Punto 4; 5,6 🡪 Punto 3; 7,8 🡪 Punto2; 9,0 🡪 Punto 1  Cualquier error o confusión que surja en la selección del punto a resolver invalida el examen |

Se dispone de un archivo con el registro de pagos, mensuales y para diferentes tipos de impuestos, de los contribuyentes a la AFIP. El archivo esta sin orden con un registro por cada pago efectuado según el contribuyente y el tipo de impuesto en el año 2019. Solo están los pagos efectivamente hechos, no hay imputación multiple en el par (contribuyente, impuesto) y pueden haber omisiones de la obligación tributaria.

Cada registro contiene:

|  |  |  |  |
| --- | --- | --- | --- |
| Mes vencimiento | CUIL Contrib. | Tipo impuesto | Pago |
| 1..12 | entero | Int | Float |

Se requiere generar una estructura en memoria ordenada por numero de contribuyente y tipo de impuesto con los pagos de cada uno de los meses del año que se hicieron efectivas las contribuciones tributarias.

1. Desarrolle la Función *Nodo\* generarEstructura(archivo, estructura)* que retorne un puntero a la estructura creada para almacenar en memoria la totalidad de los datos del archivo según los requerimientos y al final del proceso muestre los datos de pago ordenados. Declare y defina la estructura a utilizar justificando la elección.
2. Desarrolle la Función *void mayorPago( estructura)* que utilizando la estructura que considere mas idónea para resolver el punto 1 informe el numero de contribuyente que efectuo el mayor pago anual. Justifique, en un lenguaje coloquial, porque selecciono de la estructura para guardar los datos No se pide definición ni declaración de la misma
3. Desarrolle la Función *void controlDePagos( estructura)* que utilizando la estructura que considere mas idónea para resolver el punto 1 por cada contribuyente e impuesto la cantidad de meses que no efectivizo el tributo fiscal. Justifique, en un lenguaje coloquial, porque selecciono de la estructura para guardar los datos No se pide definición ni declaración de la misma
4. Desarrolle la Función *float recaudacionPorImpuesto(estructura, impuesto)* que utilizando la estructura que considere mas idónea para resolver el punto 1, reciba ademas un numero de impuesto, retorne el importe total obtenido por el pago del mismo por todos los contribujentes, elsormando además el importe total de cada uno de ellos. Justifique, en un lenguaje coloquial, porque selecciono de la estructura para guardar los datos No se pide definición ni declaración de la misma
5. Desarrolle la Función *void ordenarArchivo(archivo, estructura)* que utilizando la estructura que considere mas idónea para resolver el punto 1, reordene el archivo por contribuyente e impuesto, manteniendo la estructura original del registro por cada contribuyente. Justifique, en un lenguaje coloquial, porque selecciono de la estructura para guardar los datos No se pide definición ni declaración de la misma.