UTN.BA AyED – **Recup2 - Parcial 2** Curso: K _ _ _ _ Día: Turno: Fecha: **24-feb-2025**
Apellido: Nombre: Nro.Legajo: Nota:

Enunciado del problema a resolver. (Puntaje Total 10 (diez) sin errores. Tiempo máximo total 1h:30m).

Se evalúan los temas de: Uso de una Metodología para resolver problemas. Estructuras de Datos Dinámicas, Archivos Binarios aplicando Módulos y / o Estructuras de Datos estáticas si corresponde.

Se requiere realizar un proceso que "**Indexe el archivo de Articulos.Dat, Listar y Grabar**"; contando para ello con el siguiente archivo binario de datos:

- **Articulos.Dat**: sin orden, conteniendo:

a) Código Artículo *short*	**b)** Cantidad *short*	**c)** Descripción *str20*	**d)** Precio Unitario *float*

Se deberá considerar el siguiente bloque principal:

```
main() {   // Definir tipos, variables, inicialización y argumentos a incluir en cada invocación a las funciones.
  Abrir( _________ );  // Abre para Leer/Grabar en Articulos.Dat y Grabar solamente en IdxCodArt.Idx y en IdxDescrip.Idx.
  ActualizarArt(   );  // Aplica la constante PORCTJE a c/u. de los artículos leídos y en el mismo archivo actualiza grabando.
  ProcIdxArt(      );              // Genera ambas listas de índices recorriendo secuencialmente el archivo de Artículos.
  ListarOrdCodArt( _____________ );                    // Lista todos los Artículos ordenado por Código de Artículo.
  ListarOrdDescrip(      );                                  // Lista todos los Artículos ordenado por Descripción.
  GrabarIdxCodArt(       );        // Graba la lista de índices de Código de Artículo en archivo binario IdxArtCodArt.Idx.
  GrabarIdxDescrip( _____________ );       // Graba la lista de índices de Descripción en archivo binario IdxArtDescrip.Idx.
  Cerrar( _________ );                                                  // Los archivos indicados en Abrir.
  return 0;
} // main
// NOTA: 1: Los módulos indicados en negrita son los que se deben desarrollar.
         2: Lo indicado como ( _________ ) se debe completar con los argumentos que correspondan establecer.
```

Tabla de diseño de invocaciones y prototipos (interfaz) de funciones

Constantes a utilizar:
const float PORCTJE = 1.5;
(A) Completar en hoja aparte struct y typedef
typedef _____ *str20[ ];* ***typedef*** _____ _____ ** tListaCA;* ***typedef*** _____ _____ ** tListaD;* **struct** sArt **struct** tInfoCA **struct** tInfoD **struct** sNodoCA **struct** sNodoD
Prototipos de funciones cuyos módulos deberán desarrollarse con los argumentos indicados:
void ActualizarArt(*fstream* &Art);
void ProcIdxArt(*fstream* &Art, *tListaCA* &ListaCodArt, *tListaD* &ListaDescrip);
void ListarOrdDescrip(*fstream* &Art, *tListaD* &ListaDescrip);
void GrabarIdxCodArt(*ofstream* &IdxCodArt, *tListaCA* &*Lista*CodArt);
Prototipos de funciones que NO deberán desarrollarse pero si llamarlas con los argumentos adecuados:
void ActReg(fstream &Art, sArt rArt);
void SacarPrimerNodo(tListaCA &ListaCodArt, tInfoCA &valor);
void InsertaNodo(*tListaCA* &ListaCodArt, *tInfoCA* valor); **(1).**
void InsertaNodo(*tListaD* &ListaDescrip, *tInfoD* valor); **(2).**
void ListarOrdCodArt(*fstream* &Art, *tListaCA* &ListaCodArt); // Indicar los argumentos en el main.
void GrabarIdxDescrip(*ofstream* &IdxDescrip, *tListaD* &ListaDescrip); // " " " " " ".

Se pide:

1. **(1 punto).** Dibujar y codificar: Las estructuras de datos: indicadas en **(A) Completar struct y typedef** El dibujo debe ser**: COMPLETO, PROLIJO, SIMPLE, CLARO, y con RÓTULOS APROPIADOS.**
2. Desarrollar las siguientes funciones:
   a) **(2 puntos).** Diagramar la función **ActualizarArt** que aplica el PORCTJE acumulado a c/u. de los artículos leídos y en el mismo archivo actualiza grabando. Se invoca a **ActReg**. El proceso finaliza cuando se han leído todos los artículos.
   b) **(3 puntos).** Codificar la función **ProcIdxArt**, que genera ambas listas de índices recorriendo secuencialmente el archivo de Artículos. Debe invocar a las funciones, **InsertaNodo** según (1) e **InsertaNodo** según (2). Recordar armar los **paquetes de datos** a pasar a InsertaNodo.
   NOTA1: Recordar que en el proceso anterior se llegó al fin del archivo de artículos. ¿qué acción se debe aplicar en este otro módulo antes de volver a leer nuevamente del archivo de Artículos?.
   NOTA2: Las listas de índices se componen de un *valor clave* y su *referencia* en donde se encuentra en el archivo binario de Artículos.
   c) **(2 puntos).** Codificar la función **ListarOrdDescrip** que emite títulos apropiados y los datos de c/u. de los Artículos ordenado por Descripción. El proceso finaliza cuando se ha recorrido toda la lista sin eliminar los nodos con las descripciones de artículos incorporados. El diseño del listado es:

Listado de Artículos ordenado por Descripción

Cód.Art.	Cant.	Descripción	Pre.Unit.
9999	999	X(20)	99999.99
.	.	.	.

   d) **(2 puntos)**. Codificar o Diagramar la función **GrabarIdxCodArt** el cual graba en el archivo de *IdxCodArt.Idx* los datos de la tabla de índices de Código de Artículos. Se debe invocar a la función **SacarPrimerNodo**. El proceso finaliza cuando se han eliminado c/u. de los nodos de la lista con los códigos de artículos incorporados.

OBSERVACIONES

(1) El enunciado se complementa con explicaciones adicionales y gráficos indicados en la pizarra.
(2) **IMPORTANTE**: Resolver primero el punto 1; tiempo estimado entre 15 y 20 minutos.
(3) Establecer una muestra de datos del archivo "Articulos.Dat".
(4) En c/u. de las funciones a desarrollar se deben indicar las acciones necesarias, esto incluye ciclos, decisiones, en archivos leer o grabar, mover el puntero a la posición que corresponda, etc.
(5) Cada respuesta debe estar precedida por el rótulo de la pregunta respondida (1, 2.a, 2.b, 2.c, 2.d)
(6) Escribir el código en C++ y los diagramas en letra imprenta en forma clara, prolija, tamaño apreciable, NO escribir en los márgenes.
(7) Las salidas de los listados, deben estar dirigidas a la **consola**.
(8) Dado que el **uso de los ciclos afecta el rendimiento del proceso**, se considerará oportuno optimizar estas situaciones de las estructuras de control de programas de la **repetición**, como así también de la **selección** y la **concatenación**. **S.E.ú O.**

Folleto explicativo del problema a resolver

ESTRUCTURAS DE DATOS

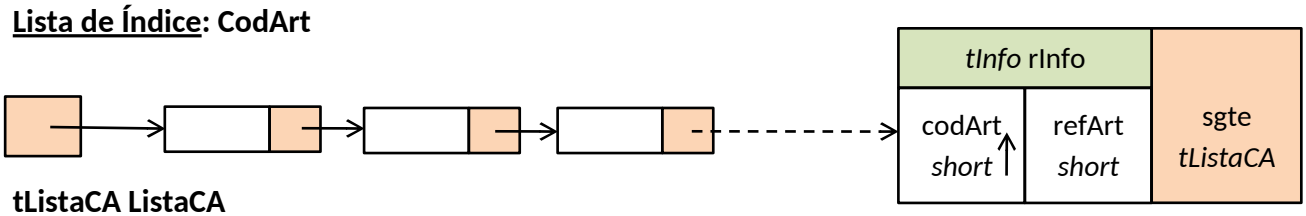
Archivo: Articulos.Dat (desordenado)

#define record **struct**

sArt			
short codArt 2b.	*short* cant 2b.	*str20* descrip 4b.	*float* preUni 4b.

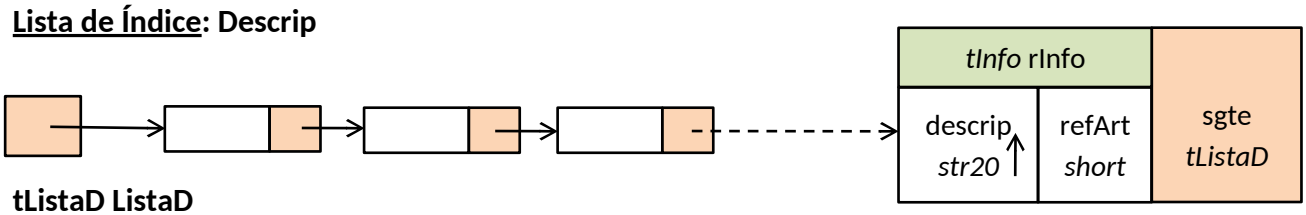
```
record sArt {
  short codArt,
        cant;
  str20  descrip;
  float   preUni;
};
```

Lista de Índice: CodArt



tListaCA ListaCA

Lista de Índice: Descrip



tListaD ListaD