

Práctico 1: Archivos

NOTA: Para trabajar con archivo en el entorno: se debe guardar cada archivo en /work/, y con un nombre que los distinga de los demás (iniciales del nombre y apellido), dado que el repositorio es compartido por todos, por ejemplo:

assign (archivo, '/work/lgomez_enteros.dat');

En este ejemplo el archivo conceptualmente se llama **enteros.dat** pero antes se agregó **/work/** porque en ese lugar están todos los archivos y luego **lgomez_** que es la inicial y el apellido del alumno para que no se “pise” con archivos de otros alumnos.

1. En cada caso realizar un procedimientos/función según corresponda:
 - a) Crear un archivo “Enteros.dat” y cargar tres valores pedidos al usuario.
 - b) Leer todos los datos que posee el archivo “Enteros.dat” e imprimirlos por pantalla. El archivo existe y posee valores.
 - c) Agregar un valor entero pedido por pantalla al final del archivo “Enteros.dat” . El archivo existe y posee valores.
 - d) Devolver el promedio de los valores de “Enteros.dat” . El archivo existe y posee valores.
 - e) Devolver el mayor de los valores de “Enteros.dat” . El archivo existe y posee valores.

2. Dadas las siguientes versiones del programa SumarArchivo:

```

program SumarArchivo;
type tarch = File of Integer;
procedure abrirArch(var arch : tarch; nombre : String; var error : Boolean);
begin
  error := false;
  assign(arch, nombre);
  {$I-} { desactiva la verificación de errores de entrada/salida (en tiempo de ejecución) }
  reset(arch); { al no estar activada la detección de errores, se puede intentar abrir
               archivos inexistentes y no se recibirán mensajes de error }
  {$I+} { activa la verificación de errores entrada/salida }
  if ioresult <> 0 then { ioresult devuelve un 0 si la última operación de entrada/salida
                     se realizó con éxito y <> 0 si hubo algún error }
    error := true;
end;
function sumarArch(var arch : tarch) : Integer;
var
  val, suma : Integer;
begin
  suma := 0;
  seek(arch, 0);
  while not eof(arch) do
    begin
      read(arch, val);
      suma := suma + val;
    end;
  sumarArch := suma;
end;
procedure escribirFinal(var arch : tarch; val : Integer);
begin
  seek(arch, fileSize(arch));
  write(arch, val);
end;
```

| Versión 1 | Versión 2 |
|---|--|
| <pre> var arch : tarch; nombre : String; error : Boolean; begin writeln('Ingresar nombre del archivo:'); readln(nombre); abrirArch(arch, nombre, error); if not error then begin escribirFinal(arch, sumarArch(arch)); close(arch); end else writeln('El archivo no existe'); end. </pre> | <pre> var orig, sum : tarch; nombre : String; error : Boolean; begin writeln('Ingresar nombre del archivo a sumar:'); readln(nombre); abrirArch(orig, nombre, error); if not error then begin abrirArch(sum, 'sumas.dat', error); if error then rewrite(sum); escribirFinal(sum, sumarArch(orig)); close(sum); close(orig); end else writeln('El archivo no existe'); end. </pre> |

- Explique brevemente qué hace cada versión del programa.
- Analice y compare el comportamiento de cada versión del programa al ejecutarla tres veces consecutivas. Suponga que el archivo utilizado, en todas las ejecuciones, es "origen.dat". El contenido inicial de origen.dat es: "4, 6, 10". El archivo "sumas.dat", al que hace referencia la versión 2, inicialmente no existe.

3. Teniendo en cuenta que los siguientes archivos son de chars, realice programas que resuelvan a los siguientes incisos:

- Dado un nombre de archivo, guardar en el archivo todos los caracteres vocales ingresados por teclado.
- Dado un nombre de archivo, generar un arreglo con los elementos ordenados ascendentemente. Suponer que el archivo contiene como máximo 100 elementos.
- Dado un nombre de archivo de origen clonarlo con otro nombre; es decir, crear otro que tenga exactamente el mismo contenido pero con un nombre distinto.

4. Teniendo en cuenta las suposiciones de los incisos a) y b) respectivamente, codifique una función que dado un número lo busque en un archivo de números y retorne la posición en la que lo encontró. Si no lo encuentra debe retornar -1.

- Suponga que los elementos del archivo se encuentran desordenados.
- Suponga que los elementos del archivo se encuentran ordenados. Tenga en cuenta que, si el archivo está ordenado, no es necesario recorrerlo completamente. Explique por qué.

5. Suponga que tiene un archivo donde cada registro es un arreglo de 5 enteros. Cada entero corresponde a la nota en la materia 1, 2, 3, 4 y 5 de los alumnos de primer año de una carrera. Cada registro del archivo corresponde a las notas de un alumno y el archivo contiene las notas de todos los alumnos de primer año de la carrera. Codifique un procedimiento (¿por qué no función?) que devuelva un arreglo con la nota promedio en cada una de las materias.

| | | | | |
|---|---|---|---|---|
| 7 | 6 | 5 | 7 | 8 |
| 3 | 9 | 8 | 5 | 4 |
| 6 | 8 | 9 | 5 | 3 |
| 5 | 6 | 5 | 7 | 8 |
| 3 | 9 | 2 | 5 | 4 |
| 6 | 8 | 2 | 4 | 3 |

Archivo

| | | | | |
|---|---|---|---|---|
| 5 | 8 | 5 | 6 | 5 |
|---|---|---|---|---|

Promedio de cada materia

6. Invierta el orden de los elementos de un archivo de caracteres.

7. Teniendo en cuenta que los siguientes archivos son de números enteros, realice procedimientos que resuelvan a los siguientes incisos:

- a. Recorra (una vez sola) un archivo de forma tal que si un número es mayor que el anterior los intercambie.
- b. Utilizando el procedimiento anterior, ordene un archivo por el método de burbujeo sobre el mismo archivo.

8. Suponga que tiene un archivo donde cada registro es un arreglo de 6 reales. Considere las cinco primeras casillas de cada arreglo son notas de materias y la última casilla el promedio. Cada promedio está inicialmente en cero. Codifique un procedimiento para actualizar el archivo de manera de completar cada promedio

| | | | | | |
|------|------|---|------|------|---|
| 7,50 | 6 | 5 | 7,50 | 8 | 0 |
| 3 | 9,50 | 8 | 5 | 4 | 0 |
| 6 | 8,50 | 9 | 5 | 3 | 0 |
| 5 | 6 | 5 | 7 | 8,50 | 0 |
| 3 | 9,50 | 2 | 9,50 | 4 | 0 |
| 8,50 | 8 | 2 | 4 | 3 | 0 |

9. Codifique un procedimiento para realizar una búsqueda binaria en un archivo (ordenado) de enteros. Compare la cantidad de accesos promedio de esta estrategia respecto a la del inciso 4.b.

10. Se tienen dos archivos con los números (enteros) de legajo de alumnos, uno posee los inscriptos en Intro II y el otro en Ciencias de la Computación I. Ambos archivos están ordenados por el nº de legajo. Confeccione un programa que genere un nuevo archivo con los nº de legajo de los alumnos que están inscriptos en ambas materias.