

Resolución de Problemas y Algoritmos

Clase 9 Resolución de Problemas con secuencias de datos



Dr. Diego R. García



Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Bahía Blanca - Argentina

Primitivas de Pascal para archivos de datos

- **assign(F,N)**: vincula F con N (nombre del archivo en SO).
- **rewrite(F)**: crea un archivo nuevo con el nombre N que está vinculado al **manejador de archivo F** (si ya existe otro archivo con ese nombre N se sobre-escribe y se pierde el viejo archivo).
- **write(F,e)**: en un archivo F creado con **rewrite**, **escribe** el valor de "e" a continuación del último elemento de F.
- **close(F)**: **cierra** el archivo vinculado al **manejador F**.
- **reset(F)**: **abre** un archivo existente de nombre N para leer, y queda preparado para leer el primer elemento.
- **read(F,e)**: **lee** un elemento del archivo F, copia el valor leído en "e" y queda preparado para leer el siguiente elemento (si existe) o queda en el fin del archivo.
- **eof(F)** (end of file – fin de archivo): **retorna TRUE** si se llegó al final de un archivo o **FALSE** en caso contrario.

Resolución de Problemas y Algoritmos

Dr. Diego R. García

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:

"Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 19/09/2019

Ejemplo propuesto para hacer en el pizarrón

Problema: crear un archivo de números reales llamado “numeros.reales” y permitir al usuario ingresar y almacenar en él tantos elementos en el archivo como quiera.

Algoritmo Cargar un archivo

Crear el archivo “numeros.reales”

Repetir

 pedir al usuario que ingrese un real

 Leer un elemento

 Almacenar en el archivo

 Preguntar si quiere cargar otro número

Hasta que no quiera cargar más

Cerrar el archivo.

Al pizarrón

fin.

Una implementación posible

Program crear_archivo_reales; // crea un archivo de reales llamado “numeros.reales” y permite al usuario ingresar y almacenar en el archivo tantos elementos como quiera.

TYPE archivo_reales = FILE OF real; // tipo nuevo creado por el constructor

VAR archi : archivo_reales; // manejador del archivo de reales

 opcion : char; num: real;

begin

 assign(archi, 'numeros.reales'); // vincula el nombre con el manejador

 rewrite(archi); // crea el archivo vacío en una carpeta del disco duro

REPEAT

 write('Ingrese un real: '); readln(num); // solicita y lee el número

 write(archi, num); // lo escribe al final del archivo

 write('¿Quiere ingresar otro? (S/N) '); readln(opción); // pide opción

UNTIL (opcion = 'N') or (opcion = 'n');

 close(archi); // el archivo queda cerrado y listo para ser usado por otros

end.

Problema: escriba un programa para copiar el contenido de “numeros.reales”, a otro archivo “numeros.reales.copia”.

```

Program incorrecto; {VERSION INCORRECTA}
TYPE archivo_reales = FILE OF real; // tipo nuevo
VAR F1,F2: archivo_reales; // manejadores de archivos
begin
  assign(F1, 'numeros.reales');
  assign(F2, 'numeros.reales.copia');
  F2:=F1;
  close(F2); close(F1);
end.
    
```

MAL

Esto no copia los archivos


Esta asignación hace que los dos manejadores se refieran al mismo archivo: “numeros.reales”

Resolución de Problemas y Algoritmos Dr. Diego R. García 5

Problema: escriba un programa para copiar el contenido de “numeros.reales”, a otro archivo “numeros.reales.copia”.

```

Program copiar; {version correcta}
TYPE archivo_reales = FILE OF real; // tipo nuevo
VAR F1,F2: archivo_reales; elemento:real;
begin // copia elemento a elemento de F1 a F2
  assign(F1, 'numeros.reales'); reset(F1); // abre F1 para lectura
  assign(F2, 'numeros.reales.copia'); rewrite(F2); // F2 para escritura
  while not eof(F1) do begin // mientras no se acabe F1
    read(F1,elemento); // leo de a un elemento de F1
    write(F2, elemento); // escribo cada elemento leído en F2
  end; // terminó de copiar elemento a elemento
  close(F1); close(F2); // cierro los dos archivos
  writeln('Copia realizada con éxito. '); readln;
end.
    
```



Resolución de Problemas y Algoritmos Dr. Diego R. García 6

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:

“Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 19/09/2019

Ejemplo

Problema: escriba un programa que busque **cuantas veces está** el elemento E (ingresado por el usuario) en el archivo “numeros.reales” el cual ya fue creado.

Algoritmo contar cuantas veces

Abrir el archivo para leer

Solicitar número a buscar

Asignar el valor inicial 0 a cantidad de veces

Repetir mientras no sea fin de archivo (EOF)

Leer un elemento del archivo

Si el elemento es el buscado

entonces incrementar cantidad de veces en uno

Cerrar el archivo.

Mostrar cantidad encontrada

fin.

Resolución de Problemas y Algoritmos

Dr. Diego R. García

7

```

Program cuantas_veces; {busca cuantas veces está un elemento ingresado por el usuario
en el archivo “numeros.reales” el cual ya fue creado.}
TYPE archivo_nums = FILE OF real; // tipo nuevo creado por el constructor
VAR numeros: archivo_nums; // manejador del archivo de reales
    buscado : real; // contendrá el nro buscado ingresado por el usuario
    leido: real; //contendrá valores que va leyendo de a uno del archivo
    encontrados: integer; // contendrá la cantidad de apariciones encontradas

begin
  assign(numeros, ‘numeros.reales’); // vincula el nombre con el manejador
  reset(numeros); // abre el archivo para leer
  write(‘Ingrese numero a buscar: ’); readln(buscado); // solicita y lee el número buscado
  encontrados:=0; // valor inicial para cantidad de encontrados
  WHILE NOT eof(numeros) DO // mientras no sea el fin del archivo números
    begin
      read(numeros, leido); // leo uno a uno los elementos del archivo
      IF leido= buscado // si el leído es el que está siendo buscado
        THEN encontrados := encontrados + 1; // incremento la cantidad de encontrados
      end; // end del while
  close(numeros); // el archivo queda cerrado y listo para ser usado por otros
  write(‘La cantidad de veces que está ‘, buscado:0:0, ‘ es: ’, encontrados); readln;
end.

```

Resolución de Problemas y Algoritmos

Dr. Diego R. García

8

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:

“Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 19/09/2019

Conceptos: Uso de analogías (el valor de la experiencia)

- Resolver un problema **por analogía** consiste en hacer uso de la solución encontrada para un problema ya resuelto antes (o parte de ella) para resolver un nuevo problema.
- De esta manera, se puede poner en práctica toda la **experiencia** obtenida en la solución de problemas previos.
- Resolver un problema por analogía consiste en encontrar una **correspondencia** entre los elementos involucrados en dos problemas: el problema cuya solución ya conocemos y el problema nuevo cuya solución buscamos.
- Usando esa correspondencia, hay que **adaptar** la solución del problema ya resuelto para encontrar la solución del nuevo.

Resolución de Problemas y Algoritmos

Dr. Diego R. García

Ejemplo

Problema: escriba un programa que busque **cuantas veces está** el elemento E (ingresado por el usuario) en una secuencia de números naturales ingresada por teclado (buffer) y usando 0 como terminador.

Algoritmo contar cuantas veces

Solicitar número a buscar

Asignar el valor inicial 0 a cantidad de veces

Leer un elemento ingresado por teclado

Repetir mientras no sea un 0

 Si el elemento es el buscado

 entonces incrementar cantidad de veces en uno

 Leer un elemento ingresado por teclado

Mostrar cantidad encontrada

fin.

Resolución de Problemas y Algoritmos

Dr. Diego R. García

10

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:

“Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 19/09/2019

Ejemplo

Problema: escriba un programa que busque **cuantas veces está** el dígito D (ingresado por el usuario) en un número.

Algoritmo contar cuantas veces

Solicitar número y el dígito a buscar

Asignar el valor inicial 0 a cantidad de veces

Repetir

 Si el último dígito es el buscado

 entonces incrementar cantidad de veces en uno

 eliminar el último dígito del número

Hasta que número es 0

Mostrar cantidad encontrada

fin.