

INTRODUCCIÓN A LA PROGRAMACIÓN

PRÁCTICA 6 Secuencias

Objetivos

- ✓ Presentar los paquetes de los que haremos uso para definir y utilizar variables de tipo secuencia.
- ✓ Estudiar la utilización de todo lo anterior en programas Pascal resolviendo problemas sencillos de recorrido que manejen secuencias del primer modelo de acceso secuencial.

2. Tipo máquina secuencial en Pascal

En el lenguaje Pascal no se encuentra predefinido el tipo de dato máquina secuencial, tal como lo hemos estudiado en clase de teoría. Para utilizar el tipo de datos *Secuencia*, tanto del primer como del segundo modelo de acceso secuencial, se han escrito una serie de unidades que debidamente utilizadas permiten usar este tipo de datos con las mismas primitivas de acceso vistas en clase de teoría.

Estas unidades se pueden descargar desde el entorno del aula virtual. Dentro de la asignatura: “Introducción a la Programación”, en “Recursos”, “Prácticas” hay un directorio llamado “Unidades de MMSS” donde se encuentran estas unidades. Dependiendo del tipo de datos de la secuencia que se desea manejar debemos utilizar una unidad diferente. Podemos manejar secuencias de enteros, de caracteres y de reales.

Cada archivo se ha nombrado con el prefijo “unitms” seguido de una letra que indica el tipo de los elementos de la secuencia: ‘c’, ‘e’ o ‘r’, para caracteres, enteros o reales respectivamente, y un ‘1’ o un ‘2’ para indicar el modelo. Por ejemplo, `unitmse2.pas` es la unidad para secuencias de enteros del segundo modelo y `unitmsc1.pas` es la unidad para secuencias de caracteres del primer modelo.

NOTA: Aunque Pascal no es sensible a la distinción entre mayúsculas y minúsculas en los identificadores del programa, el sistema de ficheros del sistema operativo sí puede serlo en los nombres de los archivos. Por esta razón, el nombre de la unidad que usemos deberemos ponerlo siempre con todas sus letras minúsculas, tal como está en los nombres de los archivos.

Las primitivas para el manejo de secuencias se nombran igual que en el libro de texto (EA, Comenzar, Avanzar, etc.) pero con un sufijo que indica el tipo de elemento y el modelo secuencial (EA_MSC1, EA_MSE2, Avanzar_MSR2, Avanzar_MSC2, etc.). A continuación mostramos tres ejemplos, uno para cada tipo de secuencia, del uso de estas unidades.

Ejemplo 1: El siguiente programa carga el contenido de un archivo de texto cuyo nombre es `entrada1.txt` en una secuencia de caracteres S. A continuación copia ésta en otra secuencia de caracteres R, sustituyendo cada carácter en minúscula por su correspondiente mayúscula y graba el contenido final de R, en un archivo de texto cuyo nombre es `salida1.txt`.

```

program  secuencias_caracteres_1;
uses
    unitmsc1;
const
    (* Distancia entre los juegos de caracteres mayúsculas y minúsculas: *)
    Distancia = ord ('A') - ord ('a');
var
    S, R : MSC1;
    c_conv : Char;
begin
    Encender_Maquina_MSC1 (S);
    Encender_Maquina_MSC1 (R);
    Cargar_Fichero_MSC1 (S, 'entrada1.txt');
    Comenzar_MSC1 (S);
    Crear_MSC1 (R);

    (* Esquema de recorrido del primer modelo *)

    while EA_MSC1 (S) <> MSC1_MarcaFin do
        begin
            if (EA_MSC1 (S) >= 'a') and (EA_MSC1 (S) <= 'z') then
                c_conv := chr (ord (EA_MSC1 (S)) + Distancia)
            else
                c_conv := EA_MSC1 (S);
                Registrar_MSC1 (R, c_conv);
                Avanzar_MSC1 (S)
            end;
            Marcar_MSC1 (R);
            Salvar_Fichero_MSC1 (R, 'salida1.txt')
        end.

```

Ejemplo 2: El siguiente programa carga el contenido de un archivo de texto cuyo nombre es `entrada2.txt` en una secuencia de enteros S. A continuación escribe por la salida estándar el contenido de esta secuencia al mismo tiempo que va registrando en otra secuencia R el cuadrado de cada número que aparece en S. Finalmente graba el contenido de R en un archivo de texto cuyo nombre es `salida2.txt`:

```

program  secuencias_enteros_1;
uses
    unitmse1;
var
    S, R : Mse1;
begin
    Encender_Maquina_Mse1 (S);
    Encender_Maquina_Mse1 (R);
    Cargar_Fichero_Mse1 (S, 'entrada2.txt');
    Comenzar_Mse1 (S);
    Crear_Mse1 (R);
    while EA_Mse1 (S) <> Mse1_MarcaFin do
        begin
            Write (EA_Mse1 (S), ', ');
            Registrar_Mse1 (R, EA_Mse1 (S) * EA_Mse1 (S));
            Avanzar_Mse1 (S)
        end;
    Marcar_Mse1 (R);
    Salvar_Fichero_Mse1 (R, 'salida2.txt')
end.

```

Ejemplo 3: El siguiente programa carga el contenido de un archivo de texto, que contiene las notas de un curso, cuyo nombre es entrada3.txt en una secuencia de reales S y muestra por pantalla el número de notas que superan el cinco.

```

program  secuencias_reales_1;
uses
    unitmsr1;

const
    Aprobado = 5.0;
var
    S : Msr1;
    cont : Integer;
begin
    Encender_Maquina_Msr1 (S);
    Cargar_Fichero_Msr1 (S, 'entrada3.txt');
    Comenzar_Msr1 (S);
    cont := 0;
    while EA_Msr1 (s) <> Msr1_MarcaFin do
        begin
            if EA_Msr1(s) >= Aprobado then cont := cont + 1;
            Avanzar_Msr1 (S)
        end;
        Write ('El número de aprobados es: ', cont)
    end.

```

Problema 1: A continuación se muestran un par de versiones de un mismo programa Pascal, que utilizan una máquina secuencial de caracteres del primer modelo de acceso secuencial:

<pre> program cuenta; uses unitmscl; var S : Msc1; cont : Integer; begin Encender_Maquina_MSC1 (S); Cargar_Fichero_Msc1(S, 'datos.txt'); Comenzar_Msc1(S); cont := 0; while Ea_Msc1(S) <> Msc1_MarcaFin do begin Write (EA_Msc1 (S), ','); cont := cont + 1; Avanzar_Msc1 (S) end; WriteLn; WriteLn ('Nro. de datos: ', cont) end.</pre>	<pre> program cuenta; uses unitmscl; var S : Msc1; cont : Integer; begin Encender_Maquina_MSC1 (S); Cargar_Fichero_Msc1(S, 'datos.txt'); Comenzar_Msc1(S); cont := 0; while Ea_Msc1(S) <> Msc1_MarcaFin do begin Avanzar_Msc1 (S); Write (EA_Msc1 (S), ','); cont := cont + 1 end; WriteLn; WriteLn ('Nro. de datos: ', cont) end.</pre>
---	---

- ✓ ¿Qué esquema algorítmico se está tratando de utilizar?
- ✓ ¿Cuál de los dos es el correcto? ¿Por qué?
- ✓ Antes de ejecutarlos intente predecir cuál sería el resultado que se produciría al compilar y ejecutar cada uno de ellos. Realice estos procesos con ambos y compare los resultados con sus hipótesis.

Problema 2. Supongamos el problema de la práctica tres, en la que realizábamos la codificación de un carácter introducido por teclado. ¿Cómo resolvería el problema si en lugar de un carácter tuviese que codificar una frase? Suponga en este caso que el texto viene dado en una secuencia.

Problema 3:

Dada una secuencia de enteros S , escriba un algoritmo que cree una nueva secuencia T formada por la secuencia de sumas parciales de S ($S_1, S_1 + S_2, S_1 + S_2 + S_3, \dots$). Por ejemplo para la secuencia de entrada: 1, 4, 5, 3, 5, 6. La secuencia de salida sería: 1, 5, 10, 13, 18, 24.

Problema 4:

Escriba un algoritmo que obtenga la intersección de dos conjuntos de enteros representados como secuencias ordenadas crecientemente.

Ejemplo:

$$\begin{aligned}
 S1 &= [-44, -21, -1, 3, 5, 7, 18, 24, 30, 33, 54, 66, 101] \\
 S2 &= [-32, -28, -21, 0, 3, 4, 6, 7, 14, 20, 24, 31, 33, 51, 66, 104, 200] \\
 S1 \cap S2 = R &= [-21, 3, 7, 24, 33, 66]
 \end{aligned}$$

Trabajo personal del alumno

Problema 1

Escriba una función que simule la evolución del marcador en un partido de tenis. Los datos de entrada son el marcador actual y qué jugador ha ganado el punto en juego. Como resultado se mostrará por pantalla el nuevo marcador.

Utilice esta función para escribir un algoritmo que simule la evolución de un partido de tenis, supuesto que la entrada es una secuencia S de caracteres: '1' y '2', que indica qué jugador ha ganado cada punto. Para cada elemento de la secuencia se indicará el nuevo marcador del partido. Se supondrá que el partido lo gana el jugador que consigue seis juegos. Por ejemplo, si $S = "112211221"$, la salida sería: "quince–nada", "treinta–nada", "treinta–quince", "iguales a treinta", "cuarenta–treinta", "juego para el jugador 1", "nada–quince", "nada–treinta", "quince–treinta".

Problema 2

La Caja de ahorros del Mediterráneo gestiona la atención a los clientes mediante un sistema automático. Dispone de un número fijo de ventanillas de atención (N , constante entera positiva conocida). Cada cliente, para ser atendido, debe pulsar un botón de turno, si hay alguna ventanilla libre, una pantalla digital mostrará en mensaje ("Pase"), en caso contrario mostrará el mensaje ("Espere por favor") y el usuario se pondrá en una cola, hasta que pueda ser atendido, en el momento que un empleado quede libre, la pantalla mostrará el mensaje ("Pase") si la cola no está vacía.

Escriba un algoritmo que muestre en pantalla, en el momento oportuno, cada uno de estos dos mensajes. La entrada del algoritmo será una secuencia de valores enteros: "1", "2", que sigue el primer modelo de acceso secuencial. El "1" indica una petición por parte de un cliente, el "2" que una ventanilla ha quedado libre. Inicialmente todas las ventanillas estarán libres.