

Programación 1
Segundo Parcial
Instituto de Computación - Facultad de Ingeniería
Julio 2020

Leer con atención

- Todos los programas o fragmentos de programas deben ser escritos en el lenguaje **Pascal** tal como fue dado en el curso. A grandes rasgos, este es el Pascal estándar con algunos agregados, a saber:
 - Utilización de **else** en la instrucción **case**.
 - Evaluación por circuito corto de las operaciones booleanas (**and** y **or**).
- En todos los problemas se evaluará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios impartidos en el curso. De esta manera se restarán puntos, entre otros conceptos, por: mala o nula indentación, mala utilización de las estructuras de control, código confuso e innecesariamente largo, programas ineficientes, utilización de variables globales, pasaje incorrecto de parámetros, etc. No obstante, por razones prácticas no exigimos que incluya comentarios en los códigos que escriba en la prueba.
- Escriba su nombre completo y cédula en todas las hojas.
- Numere todas las hojas y escriba la cantidad total de hojas.
- Escriba de un solo lado de la hoja y comience cada ejercicio en una nueva hoja.

Ejercicio 1 (24 puntos)

Dadas las siguientes declaraciones:

```
CONST N = ...; (* algún entero mayor que cero *)
      separa = ' ';
TYPE TipoFrase = Array [1..N] of Char;
```

Escribir la función `PalabraMasLarga` que, dada una frase, devuelve el largo de la palabra de mayor largo en dicha frase. Las palabras están separadas por uno o mas espacios. Además, puede haber espacios al inicio y/o al final de la frase.

```
FUNCTION PalabraMasLarga (fra : TipoFrase) : Integer;
```

Ejemplos para $N = 20$:

- Para `fra = " "`, el resultado es 0
- Para `fra = "esta es una frase "`, el resultado es 5
- Para `fra = " hay parcial hoy "`, el resultado es 7
- Para `fra = "el clima es lluvioso"`, el resultado es 8

Ejercicio 2 (30 puntos)

Dadas las siguientes declaraciones:

```
CONST MAX = ...; (* algún entero mayor que cero *)
TYPE
  Natural = 0..MAXINT;
  Secuencia = RECORD
    valores : ARRAY [1..MAX] OF Natural;
    tope : 0..MAX;
  END;
```

a) (15 pts) Escribir la función `EsSecuenciaOrdenada` que, dada una secuencia, devuelve `true` si sus valores no se repiten y están ordenados de menor a mayor, `false` en otro caso.

```
FUNCTION EsSecuenciaOrdenada (sec : Secuencia) : boolean;
```

Ejemplos para $MAX = 10$:

- Para `sec = []`, el resultado es `true`
- Para `sec = [3,5,6]`, el resultado es `true`
- Para `sec = [6,5,3]`, el resultado es `false`
- Para `sec = [3,5,5,6]`, el resultado es `false`

b) (15 pts) Escribir el procedimiento `AgregarValor` que, dados un valor y una secuencia ya ordenada de menor a mayor, inserta el nuevo valor en la ubicación que le corresponda, de tal manera que la secuencia siga ordenada tras la inserción. Asuma que el valor no pertenece a la secuencia y que la secuencia tiene espacio suficiente para almacenar el nuevo valor.

```
PROCEDURE AgregarValor (val : Natural; VAR sec : Secuencia);
```

Ejemplos para MAX = 10:

- Para val = 8 y sec = [], el resultado es sec = [8]
- Para val = 8 y sec = [3,5,6], el resultado es sec = [3,5,6,8]
- Para val = 8 y sec = [5,6,9,10], el resultado es sec = [5,6,8,9,10]

Ejercicio 3 (24 puntos)

Dadas las siguientes declaraciones:

```
TYPE
  Lista = ^Celda;
  Celda = RECORD
    elemento: Integer;
    siguiente: Lista;
  END;

  Resultado = RECORD
    CASE vacia : boolean OF
      true : ();
      false : (min: integer);
    END;
```

Escribir el procedimiento BuscarMinimo que, dada una lista de enteros, devuelve el mínimo de los valores de la lista en el campo min del parámetro resu y pone el campo discriminante en false, si la lista no es vacía. Si la lista es vacía pone el campo discriminante en true.

```
PROCEDURE BuscarMinimo (lst: Lista; VAR resu: Resultado);
```

Ejemplos:

- Para lst = [], el resultado es resu.vacia = true
- Para lst = [5], el resultado es resu.vacia = false, resu.min = 5
- Para lst = [6,2,9,4], el resultado es resu.vacia = false, resu.min = 2

Ejercicio 4 (12 pts)

Dado el siguiente programa, escribir qué despliega si la variable v1 del programa principal se carga con el dígito verificador de su propia cédula de identidad. Por ejemplo, si su cédula es 1.234.567-8, su dígito verificador es 8.

```
program alcance(input,output);

VAR v1: integer;
    v2: integer;

Function f1 (a: integer): integer;
BEGIN
  a:= a + 2;
  f1:= a;
END;

Procedure p1 (var v3: integer);
var v1 : integer;
BEGIN
  v3:= f1(v3);
  v1:=v3 * 2;
  Writeln(v1);
END;

BEGIN
  Read(v1);
  v2:= v1 + 2;
  p1(v2);
  Writeln(v1);
  Writeln(v2);
END.
```
