Examen de Programación 1
Instituto de Computación - Facultad de Ingeniería
Febrero 2017

Leer con atención

- Todos los códigos deben ser escritos en el lenguaje **Pascal** tal como fue dado en el curso. A grandes rasgos, este es el Pascal estándar con los siguientes agregados: utilización de `else` en la instrucción `case` y 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

Dadas las siguientes definiciones:

```
const N = ... ;
type
     NumN = array [1..N] of 0..99;
     Numero = record
                  case excede : Boolean of
                      true: ();
                      false: (valor : NumN)
                  end;
```

Queremos trabajar con enteros positivos representados en base 100 (en vez de base 10, como es usual). Para esto, usamos un arreglo de tipo `NumN` en el cual registramos los dígitos en base 100, que son valores entre 0 y 99, que representan cada entero. Por ejemplo, para representar el entero `9876543210`, se requieren los 5 valores (dígitos en base 100) siguientes: `10`, `32`, `54`, `76` y `98`. Dentro del arreglo, los elementos con menor índice serán los de menor valor posicional. Si el número es representable con menos de `N` valores, se completará el arreglo con ceros en las celdas de más a la derecha.

Ejemplos para N=6:

Entero	Representación
123456789876	|76|98|78|56|34|12|
90065	|65| 0| 9| 0| 0| 0|
1000	| 0|10| 0| 0| 0| 0|
0	| 0| 0| 0| 0| 0| 0|

Por otro lado, el tipo `Numero` permite indicar, a través del campo `excede`, si un entero es representable mediante el tipo `numN`, es decir, indica si es representable en base 100 con hasta `N` valores. Con `N=6`, para todos los ejemplos anteriores el campo `excede` vale `FALSE`, en cambio, para números mayores a `100^6-1`, como `1234561234567`, el campo `excede` vale `TRUE`.

Se pide: Escribir el procedimiento

```
procedure sumaNums (n1, n2: Numero; var res: Numero);
```

que devuelve la suma de `n1` y `n2`. Si alguno de los sumandos o la suma de estos exceden la posibilidad de representación para `N`, esto se debe indicar a través del campo `excede` del resultado.

Ejercicio 2

Dadas las siguientes definiciones:

```
const N = ... ;
const M = ... ;
type
     Palabra = record
                 pal : array [1..M] of Char;
                 tope : 0..M
               end;
     ConjPalabras = array [1..N] of Palabra;
     Resultado = array [1..N] of Boolean;
```

a) Escribir la función

`function reversa(p1, p2 : Palabra): Boolean;`

que recibe dos parámetros de tipo `Palabra` y devuelve `TRUE` si la palabra `p2` es reversa de `p1` y `FALSE` en caso contrario. La palabra `p2` es reversa de la palabra `p1` si la lectura de `p1` de derecha a izquierda es igual a `p2`. Por ejemplo, *ranas* es reversa de *sanar.*

Ejemplos:

```
reversa(barco, cobra) es FALSE
reversa(saeta, ateas) es TRUE
reversa(saetas, ateas) es FALSE
```

b) Escribir el procedimiento

`procedure reversasEnConj(conjP : ConjPalabras; var res: Resultado);`

que recibe como parámetro un conjunto de palabras `conjP` y devuelve el parámetro `res` con el valor `TRUE` en las celdas correspondientes a los índices de las palabras de `conjP` para las cuales existe una palabra reversa dentro de `conjP`, y `FALSE` en las celdas restantes. En el conjunto no hay palabras repetidas.

A continuación se muestra un ejemplo de `conjP` y el correspondiente resultado de aplicar el procedimiento `reversasEnConj`:

conjP	
índice	palabra
1	*saeta*
2	*barco*
3	*frase*
4	*ateas*
5	*soja*
6	*cobra*
7	*lamina*
8	*ajos*
9	*animal*
10	*cosa*

res	
índice	booleano
1	TRUE
2	FALSE
3	FALSE
4	TRUE
5	TRUE
6	FALSE
7	TRUE
8	TRUE
9	TRUE
10	FALSE

Ejercicio 3

Dado el siguiente programa, escriba cuál será su salida si la variable `x` se carga de la entrada estándar con el último dígito de su CI (antes del dígito verificador). Por ejemplo, si su CI es 1.234.567-8, el último dígito es 7:

```
program examen (input, output);
var i, x: integer;
   procedure proc (var y: integer);
   var z: integer;
      function func (x: integer): boolean;
      begin
         z := x + 2;
         func := z mod 2 = 0
      end;
   begin
      z := y + 1;
      if func(z) then y := (x + y) mod 10
      else y := (z + y) mod 10;
      writeln(z)
   end;
begin
  readln(x);
  for i := 1 to 4 do
     proc(x)
end.
```

Ejercicio 4

Dada la siguiente definición:

```
type
    ListaInt = ^CeldaListaInt;
    CeldaListaInt = record
                        elem : Integer;
                        sig : ListaInt
                    end;
```

Se pide implementar el procedimiento:

`procedure recortar (valorIni, valorFin: Integer; var l: ListaInt);`

que, dada una lista, elimina todos los elementos que se encuentran entre la primera ocurrencia del valor `valorIni` y la primera ocurrencia del valor `valorFin` (posterior a la primera ocurrencia de `valorIni`). Si `valorIni` no ocurre en la lista, la misma se mantiene incambiada. Si `valorFin` no ocurre en la lista, deben eliminarse todos los elementos a partir de `valorIni`. Las celdas que contienen `valorIni` y `valorFin` no deben ser eliminadas.

Ejemplos:

valorIni	valorFin	l (original)	l (luego de recortar)
5	8	[ ]	[ ]
5	8	[8,**5**,5,1,3,**8**,7]	[8,5,8,7]
1	17	[4,5,5,5,**1**,3,8,7]	[4,5,5,5,1]
2	8	[4,5,5,5,1,3,8,7]	[4,5,5,5,1,3,8,7]
2	2	[4,5,5,1,**2**,3,**2**,7]	[4,5,5,1,2,2,7]
2	2	[4,5,5,1,**2**,3,3,7]	[4,5,5,1,2]
4	7	[**4**,5,5,1,2,3,8,**7**]	[4,7]