Examen de Programación 1
Instituto de Computación - Facultad de Ingeniería
Julio 2018

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 en la primera la cantidad total de hojas entregadas.
- Escriba de un solo lado de la hoja y comience cada ejercicio en una nueva hoja.

Ejercicio 1

Dadas las siguientes declaraciones:

```
CONST
     MAX = ...; {algún valor > 0}
TYPE
     Arreglo = ARRAY [1..MAX] OF Char;
```

a) Escribir el procedimiento:

```
PROCEDURE indicesDigitos (arr: Arreglo; VAR izq: integer;
                                        VAR der: integer);
```

que devuelve los índices que indican el inicio (parámetro `izq`) y el fin (parámetro `der`) de la primera secuencia de dígitos consecutivos que exista dentro del arreglo. Si el arreglo no contiene ningún dígito, se debe devolver 0 en ambos parámetros (`izq` y `der`).

Ejemplos (`MAX = 5`):

arr	*izq*	*der*
`['#', '%', '7', '5', '9']`	3	5
`['2', '&', '8', '0', '!']`	1	1
`['a', 'b', '4', '8', 'c']`	3	4
`['$', '5', '3', '6', 'c']`	2	4
`['m', 'b', 'c', 'd', 'z']`	0	0
`['7', '2', '5', '4', '8']`	1	5

b) Se agrega ahora la siguiente declaración de tipos:

```
TYPE
     TipoValor = (num, car);
     NumCar = RECORD CASE queVal : TipoValor OF
                  num : (vnum : integer);
                  car : (vcar : char);
               END;
     ArregloNumCar = ARRAY [1..MAX] OF NumCar;
```

Escribir el procedimiento:

```
PROCEDURE convertirANumCar(arr : Arreglo; VAR ncArr : ArregloNumCar);
```

que convierte un arreglo de caracteres (`arr`) a un arreglo de tipo `ArregloNumCar` de manera tal que los caracteres comprendidos entre '0' y '9' se traducen al correspondiente número entero entre 0 y 9. Los otros caracteres del arreglo `arr` (no dígitos) se copian a `ncArr` como caracteres.

c) Se agrega ahora la siguiente declaración de tipo:

```
TYPE
     ArreTope = RECORD
                    arre : Arreglo;
                    tope : 0..MAX;
                END;
```

Escribir el procedimiento:

```
PROCEDURE transferirNoDigitos (arr: Arreglo, VAR arrTp : ArreTope);
```

El procedimiento debe cargar en `arrTp` los caracteres de `arr` que **no** son dígitos, manteniendo el orden en que aparecen en `arr`.

Ejemplos (`MAX = 5`):

arr	***arrTp***
`['x', 'y', '4', '7', 'c']`	`['x', 'y', 'c'] (tope = 3)`
`['c', 'd', 'f', '3', '3']`	`['c', 'd', 'f'] (tope = 3)`
`['m', '8', '4', '5', '2']`	`['m'] (tope = 1)`
`['z', '9', '3', 'b', 'c']`	`['z', 'b', 'c'] (tope = 3)`
`['2', '7', '3', '5', '3']`	`[] (tope = 0)`

Ejercicio 2

Dadas las siguientes declaraciones:

```
TYPE
     Lista = ^Celda;
     Celda = RECORD
                dato: Integer;
                sig: Lista;
             END;
```

Escribir el procedimiento:

```
PROCEDURE duplicarCeldas (VAR L: Lista);
```

El procedimiento debe duplicar cada celda de la lista. Es decir, debe generar una nueva celda por cada celda ya existente en la lista, copiar su valor y colocar la nueva celda a continuación de la celda original.

Ejemplos:

```
Para la lista [8, 3, 5], el resultado debe ser [8, 8, 3, 3, 5, 5]
Para la lista [7], el resultado debe ser [7, 7]
Para la lista [1, 1, 2] el resultado debe ser [1, 1, 1, 1, 2, 2]
Para la lista [], el resultado debe ser []
```

Ejercicio 3

Dado el siguiente programa, escribir cuál será su salida suponiendo que la variable x del programa principal 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 alcance (input,output);
var
   x: integer;
   y: integer;

function func (a: integer): integer;
begin
    x := a + 2;
    func := x * 2
end;

procedure proc1 (y: integer);
var x: integer;
begin
    x := 2;
    y := func (x) + y;
    writeLn (x);
    writeLn (y)
end;

procedure proc2 (var z: integer);
begin
    z := func (z);
    x := z * 2
end;

begin
    read (x);
    y := x + 2;
    proc1 (x);
    writeLn (y);
    proc2 (y);
    writeLn (x);
    writeLn (y)
end.
```