

Examen de Programación 1
Instituto de Computación - Facultad de Ingeniería
Julio 2022

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 (60 pts)

Dadas las siguientes declaraciones para trabajar con números, que pueden ser enteros o reales:

```
CONST
  MAX = ...; (* constante entera mayor que cero *)
TYPE
  TipoNumero = (NumEntero, NumReal);
  Numero = RECORD CASE tipo : TipoNumero OF
    NumEntero : (valorEntero : integer);
    NumReal : (valorReal : real);
  END;
  ArregloConTope = RECORD
    arre : ARRAY [1..MAX] OF Numero;
    tope : 0..MAX;
  END;
```

Parte a (10 pts)

Escribir la función *SonIguales* que, dados dos números, determina si son iguales o no. Dos números son iguales cuando son ambos del mismo tipo y además tienen el mismo valor. En cualquier otro caso, no son iguales. Por ejemplo, el real 4.0 **no** es igual al entero 4.

```
function SonIguales (n1, n2 : Numero): boolean;
```

Parte b (15 pts)

Escribir el procedimiento *SumarNumeros* que, dados dos números, devuelve la suma de ambos. Si los dos números son de tipo entero, entonces la suma también es de tipo entero. En cualquier otro caso, la suma es de tipo real.

```
procedure SumarNumeros (n1, n2 : Numero; var suma : Numero);
```

Parte c (20 pts)

Escribir la función *TodosIguales* que, dado un arreglo con tope de números, determina si los números almacenados en el mismo son todos iguales entre sí (según el criterio definido en la parte a) o no. En caso de que el arreglo con tope esté vacío, el resultado es true.

```
function TodosIguales (act : ArregloConTope) : boolean;
```

Parte d (15 pts)

Escribir el procedimiento *SumarTodos* que, dado un arreglo con tope de números, devuelve la suma de todos ellos (según el método de suma establecido en la parte b). En caso de que el arreglo con tope esté vacío, el resultado es de tipo entero y su valor es 0.

```
procedure SumarTodos (act : ArregloConTope; var suma : Numero);
```

Ejercicio 2 (25 pts)

Dadas las siguientes declaraciones para trabajar con una lista de números enteros:

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

Escribir el procedimiento *BorrarLuego* que, dados un valor y una lista de enteros, borra de la lista todos los valores que se encuentran luego de la primer ocurrencia del valor dado, liberando la memoria dinámica correspondiente. En caso de que el valor dado no se encuentre en la lista, no debe borrar ningún valor.

```
procedure BorrarLuego (valor : integer; var list : Lista);
```

Ejemplos:

- Si valor = 2 y list = [], el resultado es list = []
- Si valor = 3 y list = [4,2,3,7,3,5], el resultado es list = [4,2,3]
- Si valor = 4 y list = [4,2,3,7,3,5], el resultado es list = [4]
- Si valor = 5 y list = [4,2,3,7,3,5], el resultado es list = [4,2,3,7,3,5]
- Si valor = 8 y list = [4,2,3,7,3,5], el resultado es list = [4,2,3,7,3,5]

Ejercicio 3 (15 pts)

Dado el siguiente programa, escribir cuál será su salida cuando la variable x se carga de la entrada estándar con el **último** dígito de su propia cédula de identidad (antes del dígito verificador). Por ejemplo, si su cédula es 1.234.567-8, el último dígito es 7:

```
program ejercicio;
var x : integer;

procedure alfa (a: integer; b: integer);
var x : integer;
begin
  a := a + 3;
  b := b + 1;
  x := a - b;
  writeln (x)
end; (*alfa*)

procedure beta (var a: integer; var b: integer);
begin
  a := a + 1;
  b := b + 2;
  x := a + b;
  writeln (x);
  alfa (x, x)
end; (*beta*)

begin
  readLn (x);
  alfa (x, x);
  writeLn (x);
  beta (x, x);
  writeLn (x)
end.
```
