Segundo Parcial - Programación 1
Instituto de Computación - Facultad de Ingeniería
Julio 2021

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.
- Entregue solamente las hojas de solución escritas a lápiz.
- Escriba de un solo lado de la hoja y comience cada ejercicio en una nueva hoja.

Ejercicio 1 (30 pts)

Las siguientes declaraciones representan una cónica cerrada.

```
type
 Punto = record          TipoConica = (circunferencia,
           x,y: Real;                   elipse);
         end;

Conica = record case clase: TipoConica of
            circunferencia : (radio: Real; centro: Punto);
            elipse : (foco1, foco2 : Punto);
          end;

 TipoResul = record
                figura : TipoConica;
                enCentro : Boolean
             end;
```

parte a) (15 pts) Escribir el procedimiento `focoEnCentro` que devuelve en el parámetro `resultado` el tipo de cónica y el valor booleano `true` si la cónica tiene alguno de sus focos en el centro de ejes cartesianos (0.0, 0.0), o `false` si no. En el caso de una circunferencia, su centro es el foco.

```
  procedure focoEnCentro (conica : Conica; var resultado : TipoResul);
```

parte b) (15 pts) Dado un arreglo de cónicas

```
  const MAX = (* entero positivo *)
  type ArrConicas = array [1..MAX] of Conica;
```

Escribir la función `primeraCfa` que devuelve el índice del primer elemento del arreglo que es circunferencia o -1 si no hay ninguna

```
  function primeraCfa (conicas : ArrConicas): Integer;
```

Ejercicio 2 (20 pts)

```
type  Lista  = ^Nodo;
      Nodo = record
                valor: Integer;
                sig: Lista;
             end;
```

Escribir el siguiente procedimiento que dados una lista ordenada en forma creciente y un nuevo valor, inserta el valor en la lista manteniendo el orden

```
   procedure InsertOrdenado (valor: Integer; var L: Lista);
```

Ejemplos

lista original	valor	lista resultante
[4, 6, 8, 9, 15]	3	[3, 4. 6, 8, 9, 15]
[4, 6, 8, 9, 15]	7	[4, 6, 7, 8, 9, 15]
[4, 6, 8, 9, 15]	18	[4, 6, 8, 9, 15, 18]
[ ]	7	[7]

Ejercicio 3 (12 pts)

Escribir la salida del programa cuando **m** se carga de la entrada 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;
var m, z: Integer;
procedure proc1 (x,y : Integer; var m : Integer);
   var z : Integer;
   function fun1 (a, b : Integer) : Integer;
   begin
       if a >= 4 then fun1 := a-b
       else fun1 := b-a;
   end;
   begin
      z := x + y;
      m := fun1 (x,y);
      writeln (m, ' ', z);
   end;
begin
  readln (m); (* ultimo digito de antes del guion *);
  proc1 (m, m+1, z);
  writeln (m, ' ', z);
end.
```

Ejercicio 4 (13 pts)

Dadas las siguientes declaraciones:

```
const N = {entero positivo}
type  ArregloNum = array [1..N] of Integer;
      ArregloMul = record
                      multiplos: array[1..N] of Integer;
                      tope: 0..N;
                   end;
```

Escribir el procedimiento `indicesMultiplos` que retorna en el parámetro `resultado` los índices de los elementos del arreglo `enteros` que son múltiplos de la constante `cte`

```
   procedure indicesMultiplos (enteros: ArregloNum; cte: Integer; var resultado: ArregloMul);
```

Ejemplos para N=7

enteros	cte	resultado	
[45, 10, 23, 9, 15, 3, 0]	3	[1, 4, 5, 6, 7]	(tope = 5)
[45, 10, 23, 9, 15, 3, 6]	4	[ ]	(tope = 0)