

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

Dadas las siguientes declaraciones:

```
const N = ...;
type
  TipoFigura = (triangulo, cuadrado, rectangulo);
  Figura = record
    case tipo: TipoFigura of
      triangulo : (baseT, alturaT: real);
      cuadrado: (lado: real);
      rectangulo: (baseR, alturaR: real)
    end;
  Figuras = record
    elementos : array [1..N] of Figura;
    tope : 0..N
  end;
```

a) Escribir el procedimiento `sumasAreas` que, dado un arreglo con tope de figuras, devuelve la suma de las áreas de todos los triángulos, la suma de las áreas de todos los cuadrados y la suma de las áreas de todos los rectángulos.

```
procedure sumasAreas(figs : Figuras; var sumaT, sumaC, sumaR : Real);
```

b) Escribir la función `esMayor` que, dado un arreglo con tope de figuras y una figura, verifica si el área de la figura es estrictamente mayor que el de todas las figuras de ese tipo dentro del arreglo.

```
function esMayor(figs : Figuras; f: Figura) : Boolean;
```

Ejercicio 2

Se considera la siguiente definición de tipos:

```
type Lista = ^CeldaLista;
  CeldaLista = record
    elem : integer;
    sig : Lista;
  end;
```

Escribir el siguiente procedimiento:

```
procedure moverAlFinal(VAR l : Lista);
```

Que toma el primer elemento de la lista, si este existe, y lo coloca al final de la misma. Por ejemplo, si la lista original es [3, 10, 4, 19], la lista final será [10, 4, 19, 3].

Ejercicio 3

Dado el siguiente programa, determinar su salida cuando la variable num del programa principal se carga por teclado con el último dígito de su CI. Por ejemplo, si su CI es 1.234.567-8, el último dígito es 7.

```
Program alcance(input,output);
var
  flag : Boolean;
  num : Integer;

  function f (a:Boolean; var b:Integer):Boolean;
  begin
    b := (3 * b) div 2;
    f := (b = num) and a
  end;
  procedure p (var c :boolean; d:Integer);
    procedure q (num : Integer);
    begin
      num := num + num;
      writeln (num)
    end;
  begin
    c := not c;
    d := (d + 3) * 4 div 5;
    q(d);
    writeln (num)
  end;
Begin
  readln(num);
  num := ((num + 2) * 7) mod 15;
  flag := f(num mod 2 = 0, num);
  if flag then writeln('TRUE') else writeln('FALSE');
  writeln(num);
  p(flag, num);
  if flag then writeln('TRUE') else writeln('FALSE');
  writeln(3 * num)
End.
```

Ejercicio 4

En matemáticas, la suma de Riemann por la izquierda sirve para aproximar el valor de una integral, es decir, el área bajo una curva. La misma consiste en trazar un número finito de rectángulos de igual base dentro de un área irregular, calcular sus áreas y sumarlos.

Dada una función $f: \mathbb{R} \rightarrow \mathbb{R}$ que toma valores positivos en un intervalo $[a,b]$, y el entero N que determina la cantidad de rectángulos que cubren $[a, b]$, se pide escribir la función `areaCurva` que aproxima el área bajo la curva de f para el intervalo dado, calculando la suma de Riemann por la izquierda que se define de la siguiente forma:

$$S = \sum_{i=1}^n f(x_{i-1})(x_i - x_{i-1})$$

El encabezado de la función solicitada debe ser el siguiente:

```
function areaCurva(a, b : Real; N : Integer) : Real;
```

Asuma que la función f tiene la siguiente declaración y ya se encuentra implementada:

```
function f(x : Real) : Real;
```

Ejemplo para $n=4$:

