

Examen Programación 1

Febrero 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, algoritmos 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 (25 puntos)

Dadas las siguientes declaraciones para representar listas de caracteres:

```
type
  ListaChars = ^Celda;
  Celda = record
    letra : Char;
    sig   : ListaChars
  end;
```

Y la siguiente definición: *Un palíndromo es una palabra que si se lee de izquierda a derecha, es igual a leerla de derecha a izquierda. Ejemplos: asa, reconocer, seres, acfbbfca.*

Implementar el siguiente procedimiento:

```
procedure crearPalindromo(var lis : ListaChars);
```

Que recibe una lista de caracteres y la convierte en un palíndromo, espejando los caracteres de la lista original. Ejemplos:

Lista original	Lista modificada
<'h', 'o', 'l', 'a'>	<'h', 'o', 'l', 'a', 'a', 'l', 'o', 'h'>
<'a'>	<'a', 'a'>
<>	<>

Ejercicio 2 (20 puntos)

Escribir una función que convierta un entero positivo a su equivalente en base 2, representado como un entero que solo contiene los dígitos 0 y 1. Para convertir un entero en base 10 a su equivalente en base 2 (binario) se realizan divisiones enteras sucesivas entre 2. Cada cociente se divide entre 2 hasta que se obtiene un cociente igual a 0. Los restos de las divisiones forman el número binario, en donde el primer resto corresponde a las unidades. Asumir que no se superará el valor de MAXINT.

$$\begin{array}{r} 29 \overline{) 2} \\ 1 \quad 14 \overline{) 2} \\ \quad 0 \quad 7 \overline{) 2} \\ \qquad 1 \quad 3 \overline{) 2} \\ \qquad \qquad 1 \quad 1 \overline{) 2} \\ \qquad \qquad \qquad 1 \quad 0 \end{array} \quad 29 \Rightarrow 11101$$

Otros ejemplos: Al entero 26 le corresponde el binario (representado como entero) 11010

Al entero 4 le corresponde el binario (representado como entero) 100

Al entero 0 le corresponde el binario (representado como entero) 0

```
function decimalAbinario(numDec: Integer): Integer;
```

Ejercicio 3 (12 puntos)

Decir qué despliega el siguiente programa si se ingresa el último dígito de su cédula antes del guión (por ejemplo, si su cédula fuera 1.234.567-8, se ingresaría el 7).

```
program prog;
var a,b,c,n : Integer;
  procedure proc(var a: Integer; var b: Integer);
  var n : Integer;
  begin
    n:=b+a;
    while (n > b) do
      n := n-1;
    a := n;
    b := b + 2
  end;
  function fun(n: Integer): Integer;
  begin
    c := n + a;
    fun := c mod n
  end;
begin (* principal *)
  read(a);
  c := a;
  b := 10;
  n := 3;
  a := fun(n+1);
  writeln(b, ' ', c);
  proc(a,b);
  writeln(b, ' ', c);
  proc(b,c);
  writeln(b, ' ', c)
end.
```

Ejercicio 4 (43 puntos)

Las siguientes declaraciones representan información sobre la situación laboral de un conjunto de personas: de las que están empleadas se registra el sueldo, de las que no, se registra si están buscando empleo.

```
const
  MAX = ...; {valor mayor que 0}
type
  TipoEstado = (Empleada, Desempleada);
  TipoPersona = record
    identificador: Integer;
    case estado: TipoEstado of
      Empleada : (sueldo : Real);
      Desempleada : (buscaTrabajo : Boolean)
    end;
  TipoPersonas = record
    personas : array [1 .. MAX] of TipoPersona;
    tope : 0..MAX
  end;
```

Parte a)

Implementar una función que calcule el porcentaje de personas que están desempleadas y buscan trabajo, sobre el total de personas. Si el arreglo con tope está vacío, se debe retornar el valor 0.

```
function porcentajeBuscaTrabajo(pers: TipoPersonas): Real;
```

Parte b)

Implementar el siguiente procedimiento:

```
procedure agregarPersona(var pers: TipoPersonas, p: TipoPersona);
```

que inserta la persona dada por el parámetro *p* en el arreglo con tope *pers*, en la posición anterior a la primera persona desempleada de *pers*. Se debe mantener el orden original de los elementos del arreglo. Si no hay personas desempleadas, *p* se agrega al final. Asumir que el arreglo no está completo.