Examen de Programación 1
Instituto de Computación - Facultad de Ingeniería
Julio 2023

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`. y 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 (68 pts)

Dadas las siguientes declaraciones:

```
const
   CantTeclas = ...;      (* constante entera mayor que cero *)
type
  TipoColor = (Blanca, Negra);
  Tecla = record
             nota : char;
             color : TipoColor;
             case presionada : boolean of
                true: (intensidad : 1..10);
                false: ()
          end;
  Piano = array [1..CantTeclas] of Tecla;
  Acorde = record
              teclas : array [1..CantTeclas] of Tecla;
              tope : 0..CantTeclas
           end;
  Lista = ^Nodo;
  Nodo = record
            info : Tecla;
            sig : Lista
         end;
```

Parte a (20 pts)

Escriba el procedimiento *PrimeraBlancaPresionada* que, dado un piano, retorna en `resu` la primera tecla blanca del piano que está presionada. De no haber ninguna tecla blanca presionada, retorna en `resu` la última tecla del piano.

```
procedure PrimeraBlancaPresionada (pia : Piano; var resu : Tecla);
```

Parte b (20 pts)

Escriba el procedimiento *ObtenerAcordeIntensidad* que, dado un piano, retorna en `resu` todas las teclas del piano que están presionadas y tienen una intensidad estrictamente mayor que 5. Asuma que `resu` **no** está inicializada. De no haber ninguna tecla del piano que cumpla con las condiciones pedidas, retorna en `resu` el acorde vacío.

```
procedure ObtenerAcordeIntensidad (pia : Piano; var resu : Acorde);
```

Parte c (28 pts)

Escriba el procedimiento *BorrarPrimeraNegra* que, dada una lista de teclas, elimina la primera tecla negra de la lista y retorna true en `pudo`. En caso de no haber ninguna tecla negra, no elimina nada y retorna false en `pudo`.

```
procedure BorrarPrimeraNegra (var lis : Lista; var pudo : boolean);
```

Ejercicio 2 (20 pts)

Dada la siguiente declaración:

```
TYPE
  Digito = '0'..'9';
```

Escriba la función *NumeroContieneDigito* que, dado un entero `num` > 0, retorna true si alguna cifra de `num` coincide con el valor numérico correspondiente a `dig`, y false en caso contrario (asuma que efectivamente `num` > 0, no necesita controlarlo). Por ejemplo, si `num` = 8609 y `dig` = '6' retorna true. Si `num` = 12764 y `dig` = '0' retorna false.

```
function NumeroContieneDigito (num : integer; dig : Digito) : boolean;
```

Ejercicio 3 (12 pts)

Dado el siguiente programa, escriba cuál será su salida cuando la variable z 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, z: integer;

function Unica (x: integer) : integer;
begin
    x := x + z;
    z := z + x;
    Unica := x
end;

procedure Primero (x: integer; var y: integer);
var z: integer;
begin
    y := y + 1;
    writeLn (y);
    z := x + y;
    writeLn (z)
end;

procedure Segundo (var y, z: integer);
begin
    Primero (y, z);
    writeLn (y);
    x := Unica(y) + 2;
    writeLn (x)
end;

begin
   readLn (z);
   z := z + 1;
   Segundo (z, z)
end.
```