

# Segundo Parcial. Programación 1

## Instituto de Computación

### Julio 2024

#### Ejercicio 1 (25 puntos)

Dadas las siguientes definiciones:

```
const
  MAXCOL = ..;      { cota de columnas de una línea }

type
  { formato del texto }
  TipoFormato = ( Neg, Ita, Sub );
  Formato      = array [TipoFormato] of boolean;

  { un carácter en un texto incluye su formato }
  Caracter     = record
                    car : char;
                    fmt : Formato
                end;

  { arreglo con tope que representa a una línea }
  RangoColumna = 1..MAXCOL;
  Línea        = record
                    cars : array [RangoColumna] of Caracter;
                    tope : 0..MAXCOL
                end;

  PosibleCaracter = record case esCaracter : boolean of
                        true  : (c: Caracter);
                        false : ()
                    end;
```

#### Parte A)

Escribir la función:

```
function algunoTieneFormatoEnLinea ( tfmt : TipoFormato; ln : Línea ) : boolean;
```

que retorna true solo si algún Caracter de ln tiene el formato tfmt. En otro caso, o si ln está vacía, retorna false.

#### Parte B)

Escribir el procedimiento:

```
procedure insertarCharEnLinea ( c : Caracter; columna : RangoColumna
                               ; var ln : Línea; var pc : PosibleCaracter );
{ Precondiciones: 1 <= columna <= ln.tope + 1
                  columna + 1 <= MAXCOL }
}
```

que inserta el Caracter c (con su formato) en la columna de ln y desplaza un lugar hacia la derecha los restantes caracteres de la línea. Si (ln.tope + 1) supera MAXCOL, el carácter sobrante se retorna en pc.

#### Ejercicio 2 (10 puntos)

Se considera la siguiente definición de lista:

```
type
  Lista = ^TipoCelda;
  TipoCelda = record
    dato: char;
    sig: Lista
  end;
```

Escribir el procedimiento:

```
procedure concatenar (var l1 : Lista; l2 : Lista);
```

que recibe dos listas l1 y l2 que no comparten ninguna celda, y devuelve en l1 la lista que resulta de concatenar las listas l1 y l2 en ese orden. No deben crearse ni eliminarse celdas.

### Ejercicio 3 (16 puntos)

Se considera la siguiente definición de arreglo:

```
const MAX = ...; {valor mayor que 1}
type Cadena = array [1..MAX] of char;
```

Escribir la función:

```
function esInverso (c1, c2 : Cadena): boolean;
```

que recibe dos cadenas *c1* y *c2* y retorna *True* si y solo si *c2* es la inversa de *c1*. Una cadena es inversa a otra cuando contiene los mismos caracteres, pero en el orden contrario.

#### Ejemplo

La cadena ['h', 'o', 'l', 'a'] es inversa de ['a', 'l', 'o', 'h'].

### Ejercicio 4 (9 puntos)

Dado el siguiente programa, indicar qué despliegan las instrucciones *writeln* cuando se lee el dígito de su cédula de identidad POSTERIOR al guión. Por ejemplo, si su cédula es 1234567-8, se ingresa 8.

```
program parcial2;
var
  b, x : integer;
function func (m:integer; var n:integer): integer;
var
  b: integer;
begin
  b := m + 2;
  if n > b then
    n := m - b
  else
    n := m + b;
  func := n
end;
procedure proc (var r:integer);
begin
  r := b + r;
  writeln(r)
end;
begin
  readln(b);
  b := b + 1;
  x := func(b,b);
  proc(x);
  writeln(b);
  writeln(x)
end.
```