Exámen Final Estructuras de Datos I

Universidad Autonoma de Occidente – Cali
Depto. Operaciones y Sistemas - Facultad de Ingeniería
Prof. Luis E. Garreta U.
lgarreta@uao.edu.co

23 de mayo de 2018

- Nombre:_____
- Codigo:_____

1. (1 Punto) Realizar una función que retorne la posición del elemento menor de una lista. Un ejemplo de uso de la función es:

```
// Main
Lista l1;
l1.crear ()
l1.adicionar (66)
l1.adicionar (33)
l1.adicionar (55)
l1.adicionar (77)
l1.adicionar (22)
l1.adicionar (99)        // l1={66,33,55,77,22,99}

pos = l1.buscarMenor () // pos = 4
```

2. (2 Puntos) El algoritmo de ordenamiento recursivo MergeSort está implementado de la siguiente manera:

```
algoritmo MergeSort (lx: Lista){
  Lista l1, l2, listaResultado;

  if lx.estaVacia () == VERDAD)
    return (lx)
  else {
    n = lx.longitud ()
    mitad = n / 2
    l1 = lx.sublista (1, mitad)   // Primera mitad
    l2 = lx.sublista (mitad+1, n) // Segunda mitad
    listaResultado = mezclar (l1, l2)
    return (listaResultado)
  }
}
```

De acuerdo a este algoritmo, implemente la función sublista y la función mezclar, de acuerdo a las siguiente especificaciones:

a) La función ***sublista*** es llamada por una lista y le ingresa una posición inicial y final y retorna una nueva lista con los elementos desde la posición inicial hasta la posición final, sin alterar a la lista que la llama. Un ejemplo de la función ***sublista***:

```
// Main =
Lista l1;
l1.crear ()
l1.adicionar (66)
l1.adicionar (33)
l1.adicionar (55)
l1.adicionar (77)
l1.adicionar (22)
l1.adicionar (99) // l1 = {66,33,55,77,22,99}

Lista l2 = l1.sublista (1, 4) // l2 = {66,33,55,77}
Lista l3 = l1.sublista (5, 6) // l3 = {22,99}
```

b) La función ***mezclar*** le ingresan dos listas y las combina en una nueva lista de tal manera que esta nueva lista queda ya ordenada. Suponga que las dos listas tienen el mismo tamaño. Para su implementación USE la función ***buscarMenor*** del ejercicio anterior. Un ejemplo de uso de ***mezclar***:

```
// Main
Lista l1;
l1.crear ()
l1.adicionar (66)
l1.adicionar (33)
l1.adicionar (55)      // l1 = {66,33,55}

Lista l2;
l2.crear ();
l2.adicionar (99)
l2.adicionar (22)
l2.adicionar (77)      // l2 = {99,22,77}

Lista l3;
l3 = mezclar (l1, l2) // l3 = {22,33,55,66,77,99}
```

3. (2 puntos) Una expresión matemática se puede expresar de varias formas: una es la convencional o *infija* donde es operador está en medio de los operandos, ejemplo:

```
Notación infija:    (5 + 8) * (3 - 2)
```

Otro tipo de notación es la postfija, donde el operador está después. Por ejemplo, usando la anterior expresión, esta quedaría

```
Notación postfija:  5 8 + 3 2 - *
```

Implemente un evaluador de expresiones postfijas donde la expresión está contenida en una pila. Así

```
Pila p;
p.crear ()
p.adicionar ('*')
p.adicionar ('-')
p.adicionar ('2')
p.adicionar ('3')
p.adicionar ('+')
p.adicionar ('8')
p.adicionar ('5')

val = p.evaluar()
// val = 13
```