

Práctico 2 - Parte 1

1. Dados dos arreglos $a, b : \mathbf{array}[1..n] \text{ of } \mathbf{nat}$ se dice que a es “lexicográficamente menor” que b si existe $k \in \{1 \dots n\}$ tal que $a[k] < b[k]$, y para todo $i \in \{1 \dots k - 1\}$ se cumple $a[i] = b[i]$. En otras palabras, si en la primera posición en que a y b difieren, el valor de a es menor que el de b . También se dice que a es “lexicográficamente menor o igual” a b si a es lexicográficamente menor que b o a es igual a b .
 - (a) Escribir un algoritmo `lex_less` que recibe ambos arreglos y determina si a es lexicográficamente menor que b .
 - (b) Escribir un algoritmo `lex_less_or_equal` que recibe ambos arreglos y determina si a es lexicográficamente menor o igual a b .
 - (c) Escribir un algoritmo `lex_compare` que recibe ambos arreglos y devuelve valores en el tipo enumerado (menor, igual, mayor). ¿Cuál es el gollete (sentido, interés) de este algoritmo?
2. Dado el tipo

```

type person = tuple
    name: string
    age: nat
    weight: nat
end tuple
    
```

- (a) escribí un algoritmo que calcule la edad y peso promedio de un arreglo $a : \mathbf{array}[1..n] \text{ of } \mathbf{person}$.
 - (b) escribí un algoritmo que ordene alfabéticamente dicho arreglo.
3. Escribir un algoritmo que dada una matriz $a : \mathbf{array}[1..n, 1..m] \text{ of } \mathbf{int}$ calcule el elemento mínimo. Escribir otro algoritmo que devuelva un arreglo $\mathbf{array}[1..n]$ con el mínimo de cada fila de la matriz a .
4. Escribir un algoritmo que dadas dos matrices $a, b : \mathbf{array}[1..n, 1..m] \text{ of } \mathbf{nat}$ devuelva su suma.
5. Escribir un algoritmo que dadas dos matrices $a : \mathbf{array}[1..n, 1..m] \text{ of } \mathbf{nat}$ y $b : \mathbf{array}[1..m, 1..p] \text{ of } \mathbf{nat}$ devuelva su producto.
6. Dado un arreglo `cal2016` : $\mathbf{array}[1..52, \text{mon}.. \text{fri}, 9..17] \text{ of } \mathbf{string}$, donde el primer índice refiere a las semanas del año 2016, el segundo a los días hábiles de la semana y el tercero a las horas laborales del día, dar un algoritmo que devuelva la misma información en un arreglo con el formato modificado: $\mathbf{array}[9..17, 1..52, \text{mon}.. \text{fri}] \text{ of } \mathbf{string}$.
7. Dado un arreglo `cal` : $\mathbf{array}[2010..2020, 1..52, \text{mon}.. \text{fri}, 9..17] \text{ of } \mathbf{string}$, escribir un algoritmo que devuelva el número de horas en las que el calendario no señala ninguna actividad (la celda contiene el string vacío).
8. Escribí un algoritmo que dado dos punteros $p, q : \mathbf{pointer \ to \ int}$, intercambie los valores referidos. Escribí otro algoritmo que intercambie los valores de los punteros. Si hubiera un tercer puntero $r : \mathbf{pointer \ to \ int}$ que inicialmente sea igual a p , ¿cuál sería el valor referido por r luego de los intercambios en cada caso?
9. Escribí un algoritmo que devuelva el número (la cantidad) de nodos de una lista enlazada.
10. Escribí un algoritmo que devuelva el mínimo de una lista enlazada de enteros.
11. Escribí un algoritmo que agregue un nodo al final de una lista enlazada no vacía.
12. Escribí un algoritmo que construya una lista enlazada con los primeros n números naturales.
13. Escribí un algoritmo que libere toda la memoria utilizada por una lista enlazada.