

# Práctico 5: Programación Imperativa

## Algoritmos y Estructuras de Datos I 2do cuatrimestre 2018

1. (Algoritmo de la división) Dados dos números, hay que encontrar el cociente y el resto de la división entera entre ellos.

**Ayuda:** Para enteros  $x \geq 0$  e  $y > 0$ , el cociente  $q$  y el resto  $r$  de la división entera de  $x$  por  $y$  están caracterizados por  $x = q * y + r \wedge 0 \leq r \wedge r < y$ . Por lo tanto, debemos derivar un programa  $S$  que satisfaga

```
Const  $x, y : Int$ ;  
Var  $q, r : Int$ ;  
{ $P : x \geq 0 \wedge y > 0$ }           (precondición)  
S  
{ $Q : x = q * y + r \wedge 0 \leq r \wedge r < y$ }   (postcondición)
```

2. (Búsqueda lineal) Dada  $f : Nat \mapsto Bool$  y suponiendo  $\langle \exists n : 0 \leq n : f.n \rangle$ , especificar y derivar un programa que encuentra el mínimo natural  $x$  tal que  $f.x$ .

3. Dada la siguiente especificación:

```
Const  $N : Int, A : array [0, N) of Int, e : Int$ ;  
Var  $k : Int$ ;  
{ $P : \langle \exists i : 0 \leq i < N : A.i = e \rangle$ }  
S  
{ $Q : 0 \leq k < N \wedge \langle \forall i : 0 \leq i < k : A.i \neq e \rangle \wedge A.k = e$ }
```

Decir en palabras qué hace el programa y derivarlo.

4. Sea  $N \geq 0$ .

- a) Derivar un programa que calcule el menor entero  $x$  que satisface  $x^3 + x \geq N$ .
- b) Derivar un programa que calcule el mayor entero  $x$  que satisface  $x^3 + x \leq N$ .

**Ayuda:** En lugar de usar los cuantificadores de mínimo y máximo, escribir la postcondición como una conjunción.

5. (Suma de los elementos de un arreglo) Dado un arreglo de enteros, especificar y derivar un programa que calcule la suma de todos los elementos del arreglo.

6. Sea  $a$  un arreglo de enteros.

- a) Especificar y derivar un programa que determine si todos los elementos de  $a$  son positivos.
- b) Especificar y derivar un programa que determine si algún elemento de  $a$  es positivo.

7. Especificar y derivar un programa que calcule la cantidad de elementos pares de un arreglo de enteros.

8. Especificar y derivar un programa que, dado un arreglo de enteros, determine si los elementos del arreglo están ordenados en forma creciente.
9. Especificar y derivar un programa que, dados dos arreglos de enteros  $A$  y  $B$  del mismo tamaño  $N$ , calcule el producto punto entre los elementos de ambos arreglos.

**Ayuda:** El producto punto entre dos vectores  $\mathbf{a}$  y  $\mathbf{b}$  es

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n.$$

10. Especificar y derivar un programa que calcule la varianza de los valores almacenados en un arreglo.

**Ayuda:** La varianza de  $n$  valores  $\{x_1, \dots, x_n\}$  se define como:

$$\frac{\langle \sum_{i: 1 \leq i \leq n} (x_i - \bar{x})^2 \rangle}{n}$$

donde  $\bar{x}$  es el promedio de los valores.

11. Dado un arreglo  $A : array[0, N) \text{ of } Num$  con  $N \geq 0$ , contar cuántas veces coinciden dos elementos:

```
Const N : Int, A : array [0, N) of Int;
Var r : Int;
{P : N ≥ 0}
S
{Q : r = ⟨N i, j : 0 ≤ i < j < N : A.i = A.j⟩}
```

12. Dado un arreglo  $A : array[0, N) \text{ of } Num$  con  $N \geq 0$ , determinar si hay dos elementos que suman 8:

```
Const N : Int, A : array [0, N) of Int;
Var r : Bool;
{P : N ≥ 0}
S
{Q : r = ⟨∃ i, j : 0 ≤ i < j < N : A.i + A.j = 8⟩}
```

13. Dado un arreglo  $A : array[0, N) \text{ of } Num$  con  $N \geq 0$  determinar si alguno de sus elementos es igual a la suma de los anteriores.

14. Dado un arreglo  $A : array[0, N) \text{ of } Num$  con  $N \geq 0$  determinar si cada elemento tiene un valor igual al factorial de la posición en que se encuentra.

**Ayuda:** Que los elementos sean el factorial de la posición se puede expresar como  $\langle \forall i : 0 \leq i < N : A.i = i! \rangle$ .

15. Especificar y derivar un programa imperativo que calcule la función de Fibonacci de un número dado.

**Ayuda:** Recordar que la función de Fibonacci está definida de la siguiente manera:

$$\left| \begin{array}{l} fib : Nat \rightarrow Nat \\ \hline fib.0 \doteq 0 \\ fib.1 \doteq 1 \\ fib.(n+2) \doteq fib.n + fib.(n+1) \end{array} \right.$$

16. Dado un arreglo de enteros de al menos dos elementos, derivar un programa que calcule la máxima diferencia entre dos de sus elementos (en orden, el primero menos el segundo).

La especificación del programa es:

```
Const  $N : Int, A : array[0, N) \text{ of } Int;$   
Var  $r : Int;$   
{ $P : N \geq 2$ }  
S  
{ $Q : r = \langle \text{Max } p, q : 0 \leq p < q < N : A.p - A.q \rangle$ }
```

17. (Segmento de suma máxima) Dado un arreglo de enteros, derivar un programa que calcule la suma del segmento de suma máxima del arreglo.

La especificación del programa es:

```
Const  $N : Int, A : array[0, N) \text{ of } Int;$   
Var  $r : Int;$   
{ $P : N \geq 0$ }  
S  
{ $Q : r = \langle \text{Max } p, q : 0 \leq p \leq q \leq N : \text{sum}.p.q \rangle$   
  [ $\text{sum}.p.q = \langle \sum i : p \leq i < q : A.i \rangle$ ]}]
```

18. Dada la siguiente especificación:

```
Const  $M : Int, A : array[0, M) \text{ of } Int;$   
Var  $r : Int;$   
{ $P : M \geq 0$ }  
S  
{ $Q : r = \langle N p, q : 0 \leq p < q < M : A.p * A.q \geq 0 \rangle$ }
```

Decir en palabras qué hace el programa y derivarlo.

## Ejercicios extra

1. Especificar y derivar un programa que calcule el máximo de los primeros  $n$  elementos de un arreglo de tamaño  $N$  con  $0 < n \leq N$ . Tener en cuenta que no se pueden usar los valores  $\infty$ ,  $-\infty$ .
2. Especificar y derivar un programa que dado un número devuelva la posición a donde se encuentra en un arreglo. Si el elemento no está en el arreglo se debe devolver  $-1$ .
3. Considere el ejercicio 6.
  - a) En el programa derivado en el ejercicio 6a, se recorre todo el arreglo? Si la respuesta es afirmativa piense si esto es necesario. Si no lo es, busque una manera de derivar un programa equivalente que no recorra innecesariamente todo el arreglo.
  - b) Repita el mismo razonamiento sobre el ejercicio 6b y derive, si es necesario, el programa mejorado.
4. Derivar un programa para la siguiente especificación:

Const  $N : Int, A : array[0, N) \text{ of } Int;$

Var  $r : Bool;$

$\{P : N \geq 2\}$

S

$\{Q : r = \langle \exists p, q : 0 \leq p < q < N : A.p - A.q \leq 8 \rangle\}$

**Ayuda:** re-escribir la postcondición usando el operador de mínimo.

5. Derivar un programa para la siguiente especificación:

Const  $N : Int;$

Var  $a : array[0, N) \text{ of } Int; r : Int;$

$\{P : N \geq 2\}$

S

$\{Q : r = \langle \text{Max } p, q : 0 \leq p < q < N : (a.p - a.q)^2 \rangle\}$