

HOJA 6: Continuación de tipos abstractos (TAD Pila y TAD Cola)

Se parte de la especificación conocida de pilas y colas.

1. Diseñar una acción para visualizar el contenido de una pila.
2. Diseñar una acción que cuente el número de elementos de una pila.
3. Diseñar una acción que invierta el contenido de una pila.
4. Construir una acción que copie el contenido de una pila en otra.
5. Subalgoritmo que decida si dos pilas son o no iguales.
6. Diseñar un algoritmo que determine si una secuencia de caracteres es o no un palíndromo, utilizando sólo el tad Pila.
7. Se dice que una pila de enteros **representa un número** (entero positivo) si está formada por dígitos (números del cero al nueve). Podemos suponer que en la cima de la pila está la cifra de las unidades. Por ejemplo:

5
4
3
7

representaría el número 7345

- a) Escribir una función que dada una pila que representa un número, devuelva el número representado (en el ejemplo anterior la función debería devolver 7345).
- b) Escribir una función que dado un número entero positivo construya una pila que lo represente.
- c) Escribir una función que dadas dos pilas que representan números, construya otra que represente la suma de los números representados por las pilas dadas. Por ejemplo:

5
4
3
7

+

1
5
4

=

6
9
7
7

(**Nota:** pueden utilizarse las funciones de los apartados a) y b))

8. Determinar si un número almacenado en una pila es o no capicúa.
9. Subalgoritmo que decida si los elementos de una pila están ordenados.

10. Acción que añada un entero a una pila ordenada de manera que la pila resultante esté también ordenada.
11. Dadas P pila de enteros positivos ordenada de menor a mayor y Q pila de enteros, crear otra pila formada por los enteros de Q que se encuentran en las posiciones dadas por P. Se considera que la cima de una pila es la primera posición de la pila.
12. Combinar dos pilas de enteros ordenadas de menor a mayor en una sola pila ordenada de menor a mayor.
13. Construir un subalgoritmo que calcule la suma de todos los elementos almacenados en una cola de números reales.
14. Construir un subalgoritmo que calcule el máximo de los elementos almacenados en una cola de números reales.
15. Utilizando las operaciones básicas de colas y pilas, escribir un subalgoritmo que obtenga una cola con los elementos de otra cola en orden inverso.
16. Escribir un algoritmo que lea una secuencia de caracteres, introduciendo cada carácter en una pila a medida que se lee y añadiéndolo simultáneamente a una cola. Cuando se encuentre el final de la secuencia, el algoritmo tiene que utilizar las operaciones básicas de pilas y colas para determinar si la secuencia de caracteres es o no un palíndromo.
17. Construir un subalgoritmo que dadas dos colas c1 y c2 de enteros, cree una tercera cola c de enteros que sea el resultado de concatenar las dos colas anteriores. (Suponemos que todos los elementos de c1 se añaden a c antes que los de c2).
18. Construir un subalgoritmo que, a partir de dos colas de enteros ordenadas crecientemente, cree otra cola de enteros que contenga los elementos de las otras dos (sin repetir) y que también esté ordenada crecientemente.
19. Sea la siguiente especificación de TAD, que trata de representar polinomios en una variable con coeficientes reales:

Especificación TAD POLINOMIO

Género

Polinomio

Operaciones

función suma_polinomios(p,q:Polinomio) **devuelve** Polinomio
{Devuelve el polinomio suma de p y q}

función pol_x_cons(p:Polinomio;a:real)**devuelve** Polinomio
{Devuelve el polinomio a*p}

función pol_x_mon (p:Polinomio;n:entero) **devuelve** Polinomio

{Debe ser $n > 0$, devuelve el polinomio $x^n \cdot p$ }

Fin_especificación

a) Utilizando **exclusivamente** las funciones de la especificación anterior, escribir el cuerpo de una función con la siguiente cabecera:

```
funcion resta_polinomios(p,q:Polinomio) devuelve Polinomio  
{Devuelve el polinomio resta de p y q}
```

Supongamos ahora que se da la siguiente representación:

Tipo

```
Polinomio: cola(real);
```

Es decir, que representamos un Polinomio mediante una cola de números reales que serán los coeficientes de dicho polinomio, y de manera que el “primer” elemento de la cola es el coeficiente del término independiente. Por ejemplo, una cola formada por :

7, 0, -5, 0, 0, 2

(el “primer” elemento es el situado más a la izquierda) representaría el polinomio $7 - 5x^2 + 2x^5$.

Teniendo en cuenta esta representación:

- b) Implementar la función `pol_x_cons`.
- c) Implementar la función `suma_polinomios`.
- d) Implementar la función `pol_x_mon`.