

# Estructures de dades

## Curs 16-17

### Pràctica I

Biel Moyà

6 de març de 2017

**Data d'entrega:** 12 / 03 / 2017; 23:55 h

Aquesta pràctica consta de dos problemes. En el primer haureu d'implementar una estructura de dades usant cursors i en el segon en el que haureu d'emprar memòria dinàmica.

L'entrega constarà d'una carpeta comprimida que contindrà els dos projectes en carpetes separades. A la entrega, trobareu les guies de correcció per poder seguir un criteri comú.

## 1 Notació polonesa inversa

La notació polonesa inversa (RPN) és un mètode per representar expressions en què es col·loca el símbol d'operador després que els arguments que s'han d'operar. Va ser inventada l'any 1920 pel matemàtic polonès Jan Łukasiewicz. A finals de 1950, el filòsof australià Charles L. Hamblin el seu equip científic popularitzaren col·locar l'operador després dels operands.

Per exemple, la següent expressió RPN produirà la suma de 2 i 3, és a dir, 5:  $2\ 3\ +$ .

La notació polonesa inversa, també coneguda com a notació postfix, contrasta amb la "notació infix" de les expressions aritmètiques estàndard en el qual apareix el símbol d'operador entre els operands.

La RPN té la propietat de que els parèntesi no són necessaris per representar l'ordre d'avaluació o agrupació dels termes. Les expressions s'avaluen simplement d'esquerra a dreta i això simplifica en gran mesura el càlcul de l'expressió dins dels programes d'ordinador. Com un exemple, l'expressió aritmètica  $(3 + 4) \times 5$  pot expressar-se en RPN com de  $3\ 4\ +\ 5\ \times$ .

A la pràctica RPN es pot avaluar convenientment utilitzant una estructura de pila. La lectura de l'expressió d'esquerra a dreta, es realitzen les següents operacions:

1. Si l'element en curs és un enter, realitzar l'operació *push* d'aquest valor a la pila.
2. Si l'element en curs és un operador, cal realitzar l'operació *pop* dels dos elements de la part superior de la pila i realitzar un *push* del resultat de l'operació a la pila.

La RPN s'utilitza a Hewlett Packard i internament en alguns llenguatges de programació.

### Es demana

1. Implementar una pila genèrica de nombres utilitzant cursors. La part genèrica de la pila serà la mida d'aquesta.
2. Realitzar un programa anomenat calculadora que donada una expressió en format RPN, calculi el seu resultat. Les operacions permeses seran:  $+$ ,  $-$ ,  $x$ ,  $/$ ,  $\wedge$

L'expressió serà introduïda mitjançant la línia de comandes, els nombres seràn separats per espais.

**Nota:** Podreu trobar un exemple de l'ús de la línia de comandes en el repositori de l'assignatura.

**Nota2:** Al vostre projecte podeu afegir un README per especificar particularitats de la pràctica (símbols especials o consideracions que NO contradiguin l'enunciat)

## 2 (Re)organitzant llistes

En aquest segon exercici es demana que implementeu una llista usant memòria dinàmica, es a dir, usant punters. Es demana que:

1. Implementeu el package llista, el package ha de poder acceptar qualsevol tipus d'element.
2. Implementeu l'operació *dis\_order*. Aquesta operació **no pot crear cap node nou**.

3. Realitzar un programa anomenat *reorganitzador* que llegeixi una col·lecció d'enters separats per un espai d'un fixer de text anomenat *llista\_init* (notau que no té cap extensió) i mostri el resultat d'aplicar l'operació *dis\_order* per pantalla.

L'operació *dis\_order* realitza la següent tasca: Donada una llista enllaçada i un valor *x*, la reorganitzarà de tal manera que tots els nodes de menys valor que *x* es troben en primer lloc, a continuació, tots els nodes amb valor igual a *x*, i finalment els nodes amb valor més gran a *x*. L'ordre relatiu original dels nodes en cada un dels tres trams ha de ser preservat.

**Nota:** La part genèrica del package també necessitarà saber quines són les funcions de comparació per l'*item* que estau passant.

### 3 Qualificacions

Es puntuarà cada exercici sobre 100, després es farà la mitjana aritmètica de les dues notes. La forma de qualificar és la següent:

- Si funciona
  - Ben estructurat i llegible, nota màxima 100.
  - Mala utilització de les estructures (no control d'errors) o estructures innecessàries o hi ha codi de difícil lectura, nota màxima 70.
  - No es respecta la privacitat dels elements de l'estructura. Nota màxima 6.
- No funciona o no compleix els requeriments de l'enunciat
  - Sembla correcte però falla en determinats casos. Nota màxima 40
  - Esquema de l'algorisme incorrecte. Nota màxima 20
  - No compila. Nota màxima 0.

**Nota:** Quan parlem de codi de difícil lectura ens referim a bucles anidats de forma innecessària, codi spaghetti, codi sense cap comentari o amb massa comentaris.

**Nota2:** Suposarem que l'usuari és perfecte. Per exemple a l'exercici 1 pot introduir expressions amb un ordre incorrecte, però **no** introduirà lletres enlloc de nombres o operands no existents.