ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
Universitat Rovira i Virgili

Estructures de Dades 2019/20

Pràctica 1. Estructures lineals

Exercici 1 / 3. Pila d'enters sobre vector

En aquest exercici implementarem l'estructura de dades més senzilla: la pila. La pila és una estructura en la qual la inserció i l'esborrat d'elements es fan des d'una única posició. Només l'últim element a entrar a la pila serà el que en pugui sortir.

Especificació de la pila

CREAR () retorna PILA
Retorna un objecte de tipus PILA preparat per a fer-lo servir i que conté el valor de pila buida. Error si no es pot crear pila.

CIM (PILA P) retorna ELEM
Retorna l'element que està al capdamunt de la pila P. Suposa que pila existeix. Error si pila buida.

APILAR (PILA P, ELEM E) retorna PILA
Insereix l'element E al cim de la pila. Aquest element es converteix en el nou "Cim". Error si pila no existeix o no hi ha espai per apilar un nou element.

DESAPILAR (PILA P) retorna PILA
Esborra l'element del cim de la pila P. L'element anterior es converteix en el nou "Cim". Error si la pila no existeix o és buida.

ESBUIDA (PILA P) retorna BOOLEÀ
Retorna true si la pila no conté cap element. False otherwise. Error si pila no existeix.

ESPLENA (PILA P) retorna BOOLEÀ
True si pila plena.

DESTRUEIX (PILA P)*
Allibera els recursos ocupats per la pila.

*Depenent de la vostra implementació, aquesta funció caldrà o no.

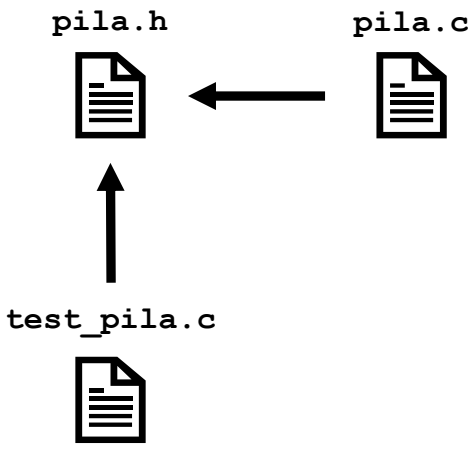
Implementació de la pila

Farem servir una implementació sobre vector dinàmic i només emmagatzemarem enters a la pila. S'ha de definir un tipus Pila (`struct`) que contindrà un vector d'enters, la capacitat màxima de la pila i una referència al nombre d'elements actuals, per poder accedir al Cim de la pila. Farem servir llenguatge C i no es poden fer servir llibreries d'estructures de dades externes. Per uniformitat, les capçaleres de les funcions han de ser exactament les següents:

```
Pila Crear(int size);
int Cim(Pila p);
void Apilar(Pila *ap, int elem);
void Desapilar(Pila *ap);
bool EsBuida(Pila p);
bool EsPlena(Pila p);
void Destruir(Pila *ap);
```

Pels booleans heu d'utilitzar la llibreria `stdbool.h`. Els paràmetres `p` i `ap` fan referència a la Pila i a un Apuntador a la Pila, respectivament. El codi s'ha d'estructurar seguint l'esquema següent:

- Declaracions de tipus i operacions a `pila.h` [INTERFACE]
- Implementació de les operacions a `pila.c` [IMPLEMENTACIÓ]
- Programa principal `test_pila.c` que comprovi que la pila funciona [CLIENT]



Aquesta uniformitat en les capçaleres i en l'estructuració del codi resultarà essencial per formar una llibreria amb totes les estructures lineals que implementem.

El programa principal `test_pila.c` serà un petit però complet joc de proves per a comprovar que efectivament l'estructura de dades funciona segons les especificacions donades.

Un cop hem comprovat que l'estructura de dades funciona correctament, implementarem una petita aplicació per donar-li ús a la nostra pila. Concretament, implementarem una calculadora RPN.

La notació RPN (Reverse Polish Notation) és un mètode d'introducció de dades que es caracteritza per utilitzar una estructura LIFO (last in first out).

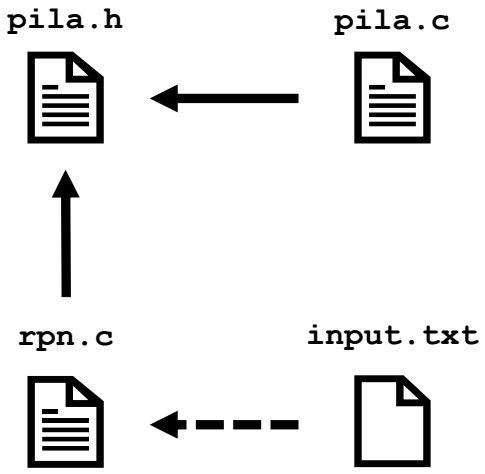
Expliquem com funciona aquesta notació amb la següent entrada:

```
7,4,3,6,+,13,-,+,-2,*,+,chs
```

Entrada	Operació	Pila	Avaluació interna
7	Apilar	7	
4	Apilar	7, 4	
3	Apilar	7, 4, 3	
6	Apilar	7, 4, 3, 6	
+	Operador suma	7, 4, 9	Desapila els dos últims valors inserits a la pila (3 i 6), els suma i apila el resultat (9).
13	Apilar	7, 4, 9, 13	
-	Operador resta	7, 4, -4	Desapila els dos últims valors inserits a la pila (9 i 13), els resta i apila el resultat (-4).
+	Operador suma	7, 0	Desapila els dos últims valors inserits a la pila (4 i -4), els suma i apila el resultat (0).
-2	Apilar	7, 0, -2	
*	Operador producte	7, 0	Desapila els dos últims valors inserits a la pila (0 i -2), els multiplica i apila el resultat (0).
+	Operador suma	7	Desapila els dos últims valors inserits a la pila (7 i 0), els suma i apila el resultat (7).
chs	Operador canvi de signe	-7	Desapila l'últim valor inserit a la pila (7), li canvia el signe i apila el resultat (-7). Ara el valor -7 és el cim i únic element de la pila, i el resultat de l'operació corresponent a l'entrada.

Per tant, podem veure que, en finalitzar tots els passos, el resultat final de l'operació és -7, i aquest valor és el que queda emmagatzemat a la pila, el qual és l'únic element d'aquesta. Observeu que, amb calculadores RPN, no fan falta els parèntesis.

Per implementar aquesta calculadora RPN crearem l'arxiu `rpn.c`, que contindrà el codi necessari per a poder, donada una entrada (arxiu `input.txt`), calcular el resultat final de l'operació:



Requisits de la calculadora:

- L'entrada es troba a l'arxiu `input.txt`, amb operands i operadors separats per comes
- Els operands són números enters (amb o sense signe)
- La pila ha de tenir capacitat per a 10 enters
- S'hauran de poder utilitzar els operadors: suma (`+`), resta (`-`), producte (`*`), divisió (`/`) i canvi de signe (`chs`). Els quatre primers operadors tenen dos arguments, el cinquè només un.
- Tant operands com resultats són enters, i les operacions segueixen les regles de C
- Es retorna error si:
    - Es divideix per 0
    - En cas de llegir un operador, si hi ha menys valors dels necessaris a la pila per poder realitzar l'operació
    - Si en finalitzar el tractament de tots els elements queden emmagatzemats més d'un element a la pila

Opcional:

- Implementar operadors addicionals: factorial, potència, mòdul, doble factorial, números d'Euler, números combinatoris, números triangulars, etc.
- Implementar operacions de manipulació de la pila: duplicació (desempila un element, i s'empila dues vegades, i.e., `x` → `x, x`), permutació (desempila dos elements, i els apila en ordre invers, i.e., `x, y` → `y, x`), cicle de tres elements (`x, y, z` → `z, x, y`), repetició (e.g., `x, 5` → `x, x, x, x, x`), etc.

- Implementar una calculadora científica RPN per a treballar amb números reals en lloc d'enters, incloent-hi les operacions més típiques: operadors aritmètics, funcions trigonomètriques, logaritmes, exponencials, potències, número pi, número e, funcions hiperbòliques, etc.

Lliurament:

- Les pràctiques són individuals
- Es fa el lliurament d'un únic arxiu `PR1.1-NOM_COGNOMS.zip` que conté:
  - Els arxius principals: `pila.h`, `pila.c`, `rpn.c`, `test_pila.c`, `input.txt`
  - Un arxiu `README.txt` on s'expliquen aquells detalls d'implementació o funcionament que creieu important remarcar, i la llista detallada d'opcions incorporades (en cas que n'hi hagi)
  - Altres arxius (`*.c`, `*.h`) que facin falta per a la implementació d'opcions (e.g., una pila de reals)