Análisis de Algoritmos y Estructuras de Datos

Práctica 4: Uso del TAD Pila

Versión 2.1

main (menu)
oper_pilas
e_s_pilas
pila

Pasos a seguir

1. Implemente el TAD *Pila* utilizando las representaciones pseudoestática y dinámica.
2. Escriba módulos que contengan las implementaciones de los subprogramas demandados en cada problema.
3. Para cada uno de los problemas, escriba un programa de prueba donde se realicen las llamadas a los subprogramas, comprobando el resultado de salida.

Ejercicios

1. Escriba una función que determine y devuelva si una secuencia de caracteres de entrada es de la forma $X\&Y$, donde X es una cadena de caracteres e Y es la cadena inversa, separadas por el carácter '&'. Recibe cadena y devuelve bool

2. Escriba una función que determine y devuelva si una secuencia de caracteres de entrada es de la forma $A\#B\#C\#D\ldots$, donde A, B, C, D,... son de la forma $X\&Y$, que a su vez están separados por el caráter '#'. Utilizamos la funcion del eje1 Recibe cadena, devuelve bool

3. Escriba una función que dados dos elementos a y b y una pila, invierta la secuencia delimitada por ambos dentro de ella.

Dado que la pila puede tener elementos repetidos, se invertirá la secuencia delimitada por la primera ocurrencia de a y de b. Primera ocurrencia de b posterior a a

Ejemplos: a=1, b=2 El tope es 4 en el primer ejemplo

Recibe dos elementos y una pila, no devuelve nada porque la pila modificada se utiliza con punteros

Pila original: 4789918679624893210
Pila final: 4789926976814893210

Pila original: 342342134219102
Pila final: 342342431219102

Pila original: 3413413424223
Pila final: 3424314314223

4. Implemente una función que dado un fichero de texto, que almacena dos enteros de longitud arbitraria en las dos primeras líneas del fichero, lo modifique añadiendo una tercera línea que contenga la suma de ambos números. Por ejemplo:

Fichero antes	Fichero después
4231490870921421	4231490870921421
2341200001	2341200001
	4231493212121422

Para la E/S en el fichero deberá usar la clase `fstream` de la biblioteca estándar de C++, declarada en la cabecera del mismo nombre. Puede seguir el esquema siguiente para leer y escribir los números en el fichero de texto.

```
#include <fstream>
#include <string>

string nombre_f; // nombre del fichero de datos

// Obtener nombre del fichero
// ...

fstream f(nombre_f); // Abrir el fichero en modo lectura/escritura y
                     // asociarlo al flujo f.
Pila<int> P, Q, R;

f >> P >> Q;              // Extraer de f los números (pilas) P y Q
// Calcular el resultado en la pila R
// ...
f << R;                          // Insertar en f el número (pila) R
f.close();                       // Cerrar el fichero asociado a f
```

Para compilar el código anterior sin errores es necesario definir previamente la sobrecarga de los operadores de extracción e inserción de pilas en flujos de ficheros.

```
// Apila en P los valores números de los dígitos
// extraídos del flujo de entrada fe
fstream& operator >>(fstream& fe, Pila<int>& P)
{
   char cifra;

   while (fe.get( cifra ) && cifra != '\n') // leer caracteres hasta fin de línea
      P.push(cifra -= '0'); // convierte un dígito en su valor numérico
   return fe;
}

// Inserta en el flujo de salida fs la pila P
fstream& operator <<(fstream& fs, Pila<int> P)
{
   while (!P.vacia()) {
```

```
        fs << P.tope();
        P.pop();
    }
    fs << endl;
    return fs;
}
```

5. En las operaciones de entrada y salida de datos es muy frecuente que los programas ofrezcan al usuario la posibilidad de editar los datos. Esto ocurre especialmente cuando los datos consisten en una línea de texto. En este caso el usuario puede introducir y modificar los caracteres en modo interactivo pulsando teclas que le permiten mover el cursor por la línea e insertar y suprimir caracteres. El modo de operación habitual es el siguiente:

- El cursor se puede situar en cualquier posición desde el primer carácter de la línea hasta la posición siguiente al último carácter de la línea.
- Cada vez que se introduce un carácter, el cursor avanza a la siguiente posición a la derecha.
- En modo inserción los caracteres que están en la posición del cursor y siguientes se desplazarán para dejar sitio al nuevo.
- En modo sobreescritura se sustituye el carácter de la posición del cursor por el nuevo.
- En ambos modos de escritura, si el cursor está en la última posición de la línea, simplemente se añaden al texto los caracteres introducidos y el cursor continúa en la última posición.

a) Defina una estructura de datos basada en el TAD *Pila* para representar una línea de texto de cualquier longitud. BASADA - dinamica

b) Escriba funciones que realicen con la estructura definida las siguientes operaciones:

- avanzar el cursor una posición
- retrasar el cursor una posición
- ir al final de la línea
- ir al principio de la línea
- borrar el carácter de la posición del cursor
- borrar el carácter anterior al cursor
- insertar un carácter en la posición del cursor
- sobreescribir el carácter del cursor

Utilizo dos pilas

6. Se desea implementar una función que simule el juego del solitario de las dos cartas. Los elementos que intervienen en este juego son:

- Mazo: Contiene (boca abajo) las cartas por colocar.
- Montón de descartes: Contiene las cartas (boca arriba) en proceso de colocación.
- Bases: Cuatro, una por palo, en las que se van colocando las cartas (boca arriba) de cada palo, en orden creciente.
- Objetivo: Colocar la totalidad de las cartas sobre sus bases correspondientes.

Desarrollo del juego: Se van tomando las cartas del mazo, por parejas, y se van colocando en el montón de descartes, de forma que la carta extraída en primer lugar quede por debajo de la segunda en el montón de descartes (es decir, que la pareja que se saca se le da la vuelta, quedando invertido su orden). Si el número de cartas del mazo es impar, en la última extracción sólo se tomará una carta.

Tras llevar dos cartas al montón de descartes, se accede a la carta de la cima de este montón (la de arriba) y se intenta colocar sobre la base que le corresponda. Se continúa pasando cartas del montón de descartes a sus bases correspondientes mientras sea posible. En este instante, se vuelven a extraer dos cartas del mazo y a llevarlas al montón de descartes, iniciándose de nuevo el ciclo.

En todo momento sólo es accesible la carta situada en la parte superior de cada uno de los elementos del juego.

Evidentemente, al comienzo del juego las bases estarán vacías y sólo podrán colocarse los ases, en caso de que las extracciones del mazo así lo permitan.

Cuando se agoten las cartas que hay en el mazo, se vuelven las cartas del montón de descartes al mazo, sin barajar (simplemente, se les da la vuelta y se devuelven al mazo, con lo que el orden se invierte), y se repite de nuevo el proceso de extracción de parejas del mazo.

El juego termina con éxito si se consigue situar todo el mazo sobre las bases, o con fracaso cuando se realicen todas las extracciones del mazo sin haber conseguido situar ninguna carta sobre las bases.

Se pide:

a) Diseñe estructuras de datos adecuadas para representar los elementos del juego definiendo los tipos `tCarta, tBase, tMazo, tMonton`.

b) Suponiendo que las cartas ya están barajadas y almacenadas en un vector de tipo `tCarta`, implemente una función que simule este juego y a su finalización devuelva si ha acabado con éxito o con fracaso, en cuyo caso deberá devolver también el valor de la última carta colocada en cada base.

Nota: La baraja española se compone de un total de 40 cartas repartidas en diez figuras (`AS, DOS, TRES, CUATRO, CINCO, SEIS, SIETE, SOTA, CABALLO, REY`) y cuatro palos (`OROS, COPAS, ESPADAS, BASTOS`).