



UNIVERSIDAD
DE MÁLAGA

| uma.es

Dpto. de Lenguajes y Ciencias de la
Computación

Programación de Sistemas y Concurrencia

Examen Junio 2019

APELLIDOS _____ NOMBRE _____

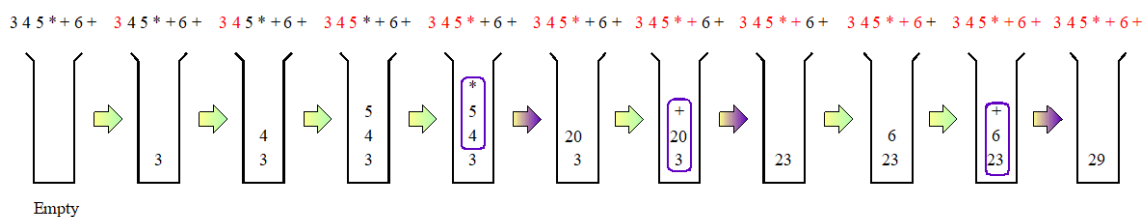
DNI _____ ORDENADOR _____ GRUPO/TITULACIÓN _____

Bloque de C (3 ptos)

Una calculadora en Notación Polaca Inversa (*Reverse Polish Notation*, RPN) permite hacer cálculos aritméticos de expresiones utilizando notación posfija. De esta manera, una expresión como $3+4*5+6$ se representaría en RPN de la siguiente manera:

3 4 5 * + 6 +

Para calcular los resultados se utiliza una pila en la que se introducen los operandos (números). Tan pronto como se lee un operador (sólo los operadores binarios +, -, * y /), éste toma como operandos a los dos números que se encuentren en la cima de la pila, los elimina y los sustituye por el resultado de operar con ellos. Como ejemplo, los cálculos anteriores producen que la pila evolucione de la siguiente manera:



donde el resultado final es el último operando que queda en la pila al final, o sea, 29. Es importante darse cuenta de que, aunque en el dibujo los operadores aparecen en la pila, estos en realidad no se insertan sino que se utilizan para evaluar los dos últimos elementos de la pila, e insertar el resultado.

Deben implementarse las siguientes funciones en el fichero fuente Stack.c:

```
// (0,1 ptos)
//Crea una pila vacía.
T_Stack create();

// (0,2 ptos)
// Devuelve true si la pila está vacía y false, en caso contrario.
int isEmpty(T_Stack q);

// (0,4 ptos)
// Inserta un número en la pila.
void push(T_Stack * pq, int operando);

// (0,6 ptos)
// Realiza la operación indicada por el operador sobre los dos
// ultimos elementos de la cima e inserta
// el resultado de la operación.
// Devuelve true si todo va bien o false en caso contrario.
int pushOperator(T_Stack * pq, char operator);
```

```
// (0,5 ptos)
// Devuelve en dato el número que hay en la cima de pila
// y lo elimina de ésta.
// Devuelve true si todo va bien o false en caso contrario.
int pop(T_Stack * pq, int * dato);

// (0,5 ptos)
// Libera la memoria de la pila y la deja vacía.
void destroy(T_Stack * pq);
```

Además, en el fichero RPN.c debe implementarse la siguiente función:

```
// (0,7 ptos)
int process(char * filename)
```

Esta función toma como parámetro el nombre de un fichero de texto que contiene una expresión en RPN (cada línea contiene sólo un operador o un operando con, a lo sumo, 5 caracteres) y devuelve el resultado de evaluarla (como el fichero es de texto, se puede ver el formato con cualquier editor). Se proporciona el fichero `source.calc` con un ejemplo de contenido de dicho tipo de ficheros de entrada.

Para facilitar la implementación de esta función, se proporcionan los siguientes métodos de ayuda:

```
int isOperator(char * text);
```

que devuelve un número distinto de 0, si es un operador

```
int text2Int(char * text);
```

que convierte un texto formado por dígitos en su equivalente numérico.

Anexo. Los prototipos de las funciones de lectura y escritura binaria en ficheros de la biblioteca `<stdio.h>` son los siguientes (se dan por conocidos los prototipos de las funciones de `<stdlib.h>` que necesites, como `free` o `malloc`):

```
FILE *fopen(const char *path, const char *mode);
```

Abre el fichero especificado en el modo indicado ("rb" para lectura binaria y "wb" para escritura binaria). Devuelve un puntero al manejador del fichero en caso de éxito y NULL en caso de error.

```
int fclose(FILE *fp);
```

Guarda el contenido del buffer y cierra el fichero especificado. Devuelve 0 en caso de éxito y -1 en caso de error.

```
char *fgets(char *str, int num, FILE *stream);
```

Lee del fichero apuntado por `stream` como máximo `num-1` caracteres o hasta que se alcanza una nueva línea o el final del fichero y devuelve la cadena de caracteres correspondiente.