

## Práctica 5. Repaso: funciones y pila

El objetivo de esta práctica es demostrar que se entienden los conceptos trabajados hasta ahora, como el uso de funciones anidadas y el uso correcto de la pila. Para ello, crearemos una nueva función que hará uso de las funciones creadas en anteriores prácticas (`itoa` y `atoi`), además de otras funciones de las cuales solo conoceremos su interfaz pública, desconociendo por completo su implementación interna.

La función creada se llamará `logaritmos`, y a partir una cadena de caracteres que representa un número en decimal, calculará el logaritmo en base 2 y el logaritmo en base 10 de este número, y devolverá una cadena de caracteres con los dos logaritmos concatenados, separados por un espacio.

1. Escriba una función llamada `logaritmos` con dos parámetros: `$a0`, que será la dirección de la cadena de entrada donde está el número expresado en decimal, y `$a1`, que será la dirección de la cadena de salida.

La función `logaritmos` utilizará internamente las siguientes funciones para realizar su trabajo:

- `atoi`: función implementada en la Práctica 4. La utilizaremos para convertir la cadena de caracteres a un entero de 32 bits en complemento a dos. Si `atoi` no devuelve error, consideraremos que la cadena es un entero válido y calcularemos sus logaritmos. En caso contrario,
- `log2`: función suministrada por Tablon para calcular el logaritmo en base 2 de un entero. Como en esta práctica vamos a trabajar solo con números enteros, el logaritmo se redondeará por truncamiento. La especificación de `log2` es la siguiente:
  - `$a0`: Número entero del cuál calcularemos su logaritmo en base 2.
  - `$v0`: Resultado de calcular el logaritmo en base 2 de `$a0`.
  - `$v1`: Código de error. Devuelve 0 si se puede calcular el logaritmo (números positivos), 1 en caso contrario (cero y números negativos).
- `log10`: función suministrada por Tablon para calcular el logaritmo en base 10 de un entero. Igual que en `log2`, el logaritmo se redondeará por truncamiento. Su especificación es idéntica a la de `log2`.
- `itoa`: función implementada en la Práctica 3. La utilizaremos para convertir los resultados de las funciones `log2` y `log10` a cadenas de caracteres.

Como podemos ver, la función `logaritmos` puede tener una salida errónea, ya sea debido a que la cadena de entrada no representa a un número entero, o bien porque el número no tiene logaritmo. Por ello, la función devolverá en el registro `$v0` un código de error, con los siguientes valores:

- 0: Todo es correcto.
- 1: Carácter incorrecto.
- 2: Número demasiado grande.
- 3: No existe logaritmo.

Los códigos de error distintos de 0 invalidan el resultado de la cadena de caracteres apuntada por el registro `$a1`.

### Ejemplos:

Cadena apuntada por <code>\$a0</code>	Cadena apuntada por <code>\$a1</code>	Valor devuelto en <code>\$v0</code>
"9"	"3 0"	0
"123"	"6 2"	0
" +123"	"6 2"	0
"1050000"	"20 6"	0
"A"	Indiferente	1
"2000000000000000"	Indiferente	2
" -1"	Indiferente	3
" -12A3"	Indiferente	3

## EVALUACIÓN Y ENTREGA DE LA PRÁCTICA

La práctica será evaluada con la herramienta Tablon, disponible en <http://tablon-aoc.infor.uva.es/>.

Para poder evaluar la práctica, el fichero `.asm` enviado al Tablon debe cumplir los siguientes requisitos:

- Debe contener la funciones `itoa` y `atoi` implementadas por el grupo en la anteriores prácticas.
- Debe contener la función `logaritmos`, así como otras funciones auxiliares creadas por el grupo. .

Una vez enviado el código, Tablon realizará una serie pruebas, testeando diferentes valores de entrada. Una vez finalizada la batería de pruebas, se podrá consultar el Tablon el número de pruebas superadas y el porcentaje de pruebas completado, además de otra información relevante, como el número de instrucciones ejecutadas, el número de líneas de código de la función y el número de excepciones en tiempo de ejecución. Estas métricas sirven para elaborar el ranking y desempatar en caso de empate en el número de casos de prueba superados. Sin embargo, **la calificación de la práctica dependiente del Tablon dependerá exclusivamente del número pruebas superadas**. El resto de métricas y el ranking no se tendrá en cuenta para la evaluación.

Finalmente, se debe entregar en el Aula Virtual el fichero `.asm` que obtuviera el mejor resultado en el ranking de la herramienta Tablon. El nombre del fichero para el Aula Virtual será el número de *request* del Tablon con 5 cifras y la extensión `.asm`

La cola para enviar esta práctica al Tablon es `lb_logs`.

Comando: `python ./client -u usuario -x password -q lb_logs NombrePrograma`

Para mandar pruebas se puede emplear la cola `mars_logs`.

Comando: `python ./client -u usuario -x password -q mars_logs NombrePrograma -- A`  
donde *A* es la cadena que representa en decimal el número del que se quieren obtener los logaritmos.

En el Tablon también aparecerá el número de ciclos necesarios para la ejecución del programa, así como el CPI. Estos datos corresponden a un procesador segmentado de 5 etapas con anticipación completa, saltos resueltos en la etapa ID y predictor de salto no tomado. En esta práctica esos datos no se evaluarán, pero sí se tendrán en cuenta en el trabajo práctico posterior.

**NOTA:** La calificación no depende exclusivamente del resultado del Tablon. El código también se revisará manualmente para evaluar su calidad y proporcionar *feedback* a los alumnos en las siguientes prácticas.

**Fecha límite de entrega en el Aula Virtual:** 31 de octubre a las 23:55 h.

**No se evaluarán las entregas del Tablon que no se hayan depositado dentro del plazo en el Aula Virtual.**