

Programa la siguiente función:

```
unsigned long long dupMasLargo(unsigned long long x);
```

Esta función calcula el duplicado más largo que aparece al final de x (es decir hacia los bits menos significativos). Revise el ejemplo de más abajo. El duplicado más largo es una secuencia de bits s tal que la secuencia ss es un sufijo de x (los bits de x son idénticos a zss para algún z). Esta función debe retornar un entero de 64 bits sin signo en donde los 32 bits más significativos son el tamaño de la secuencia s y los 32 bits menos significativos la secuencia s . En el siguiente ejemplo de uso la notación *0b...* expresa números en base 2, aunque no es parte del estándar de C:

```
unsigned long long res=
    dupMasLargo(0b101100100100100100100);
unsigned sec= res; // 0b100100100 (trunca res a 32 bits)
                // la secuencia en un entero sin signo de 32 bits
int tam= res >> 32; // 9: el tamaño de la secuencia en un entero de 32 bits
```

Restricción: Ud. no puede usar los operadores de multiplicación, división o módulo ($*$ / $\%$). Use eficientemente los operadores de bits.

Instrucciones

Baje *t1.zip* de U-cursos y descomprímalo. El directorio *T1* contiene los archivos (a) *test-t1.c*, que prueba si su tarea funciona y la compara con una implementación que usa divisiones y restos, (b) *t1.h* que incluye el encabezado de la función pedida, y (c) *Makefile* que le servirá para compilar su tarea. Ud. debe programar la función *dupMasLargo* en el archivo *t1.c*. El archivo *t1.c.plantilla* muestra algunos *includes* que Ud. debe agregar en *t1.c*. Compile su tarea con el comando *make* sin parámetros. Depure su tarea con el comando *ddd test-t1*.

Una vez que su tarea funcione correctamente, complete los siguientes pasos en un sistema operativo Linux:

- Use el comando *cd* para establecer *T1* como carpeta de trabajo.
- Cree un archivo con nombre *resultados.txt* y copie y pegue la salida de:

```
uname -a ; cat /etc/*release
```

- Pruebe su tarea compilada para depuración con el comando:

```
make test-g
```

- La salida de este comando debe terminar con el mensaje de

“Felicitaciones ...” y la compilación no debe arrojar errores o *warnings*. De lo contrario su tarea será rechazada.

- Añada al archivo *resultados.txt*: un título que diga “*depuración activada*”, luego el tiempo que toma el programa en calcular el duplicado más largo con divisiones y restos, el tiempo que toma con su función y el *speed up*. Estos valores son reportados por *test-t1*.
- Pruebe su tarea optimizada con la opción *-O* de gcc, con el comando:
make test-O
- La salida de este comando debe terminar también con el mensaje de “Felicitaciones ...” y la compilación no debe arrojar errores o *warnings*. De lo contrario su tarea será rechazada.
- Añada al archivo *resultados.txt*: un título que diga “*optimización activada*”, luego el tiempo que toma el programa en calcular el duplicado más largo con divisiones y restos, el tiempo que toma con su función y el *speed up* reportado.
- Opcionalmente si su Linux puede correr programas de 32 bits pruebe su tarea optimizada para máquinas de 32 bits con el comando: *make test-m32* (no será evaluado, no funciona bajo WSL 1)
- Opcionalmente añada el reporte de los resultados para 32 bits al archivo *resultados.txt* en el mismo formato que para la compilación con depuración u optimización (no será evaluado).

Pruebe de preferencia su tarea con la distribución Debian de Linux, como por ejemplo: (i) Debian para Windows bajo WSL 1 o WSL 2, (ii) Debian 10 bajo VirtualBox. Si no tiene acceso a Debian puede usar otras distribuciones de Linux como Ubuntu, CentOS, Kali, etc. En el resto de las tareas será obligatorio probar con Debian.

Entrega

Ud. solo debe entregar los archivos *t1.c* y *resultados.txt* en el formato *.zip* por medio de U-cursos. Además se descontará medio punto por día de atraso. No se consideran los días de vacaciones, sábado, domingo o festivos.