

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 todavía 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
```

Restricciones:

- Ud. no puede usar los operadores de multiplicación, división o módulo (* / %). Use los operadores de bits eficientemente.
- Se descontará medio punto por no usar el estilo de indentación de Kernighan como se explica en [esta sección](#) de los apuntes.
- El estándar de C no especifica el resultado para desplazamientos mayores o iguales al tamaño del operando. *Sanitize* (rechaza el desplazamiento $x \ll nbits$ cuando $nbits$ es mayor o superior a la cantidad de bits de x).

Instrucciones

Baje *t1.zip* de U-cursos y descomprímalo. El directorio *T1* contiene entre otros archivos (a) *test-dup.c* que prueba si su tarea funciona y compara su eficiencia con la solución del profesor, (b) *prof.ref-x86_64* y *prof.ref-aarch64* con los binarios ejecutables de la solución del profesor, (c) *dup.h* que incluye el encabezado de la función pedida, y (d) *Makefile* que le servirá para compilar y ejecutar su tarea. Ud. debe programar la función *dupMasLargo* en el archivo *dup.c*.

Pruebe su tarea bajo Debian 12 nativo o virtualizado con VirtualBox, Vmware, QEmu o WSL 2. **Ejecute el comando *make* sin parámetros.** Le mostrará las opciones que tiene para compilar su tarea. Estos son los requerimientos para aprobar su tarea:

- *make run* debe felicitarlo por aprobar este modo de ejecución. Su solución no debe ser 80% más lenta que la solución del profesor.
- *make run-g* debe felicitarlo.
- *make run-san* debe felicitarlo y no reportar ningún problema como por ejemplo desplazamientos indefinidos.

Cuando pruebe su tarea con *make run* asegúrese que su computador esté configurado en modo alto rendimiento y que no estén corriendo otros procesos intensivos en uso de CPU al mismo tiempo. De otro modo podría no lograr la eficiencia solicitada.

Entrega

Ud. solo debe entregar por medio de U-cursos el archivo *dup.zip* generado por el comando *make zip*. **A continuación es muy importante que descargue de U-cursos el mismo archivo que subió, luego descargue nuevamente los archivos adjuntos y vuelva a probar la tarea tal cual como la entregó.** Esto es para evitar que Ud. reciba un 1.0 en su tarea porque entregó los archivos equivocados. Créame, sucede a menudo por ahorrarse esta verificación. Se descontará medio punto por día de atraso. No se consideran los días de receso, sábados, domingos o festivos.