

Programa la función *sumaVecs* con el siguiente encabezado:

```
unsigned long long sumaVecs(unsigned long long vx,
                             unsigned long long vy);
```

El parámetro *vx* corresponde a un vector de 8 enteros de 8 bits sin signo. El vector se representa mediante un solo entero de 64 bits. Cada elemento del vector ocupa 8 bits de *vx*. Lo mismo con el parámetro *vy*. Si los bits de *vx* son  $vx_{63}...vx_0$ , para efectos de esta pregunta nos referiremos al *k*-ésimo entero de *vx* como  $vx[k]$  y corresponde a los bits  $vx_{(k+1)*8-1} ... vx_{k*8}$ , con  $k = 0...7$ . Por ejemplo si *vx* es 0xff016700a1000203,  $vx[0]$  es 3 y  $vx[7]$  es 0xff (255). La función *sumaVecs* debe retornar un vector con la suma de los vectores *vx* y *vy*. Es decir el resultado *vz* es tal que  $vz[k] = vx[k] + vy[k]$ . Ejemplo de uso:

```
Vector vx = 0xff 01 67 01 a1 00 00 03;  
Vector vy = 0x01 09 03 ff 1f 00 00 ff;  
Vector vz = sumaVecs(vx, vy);  
//      vz = 0x00 0a 6a 00 c0 00 00 02
```

Observe que al sumar los números marcados en negritas en *vx* y *vy*, se excede el rango representable en 8 bits. En ningún caso este desborde debe afectar el resultado de la suma de la siguiente columna (marcada en negritas en *vz*). Es decir el resultado de calcular  $vx[0] + vy[0]$  ( $0x03 + 0xff$ ) se trunca a 8 bits y se almacena en  $vz[0]$ , sin afectar el resultado de  $vz[1]$  que debe ser 0 en el ejemplo. Por esta razón no sirve calcular  $vz = vx + vy$ , porque así  $vz[1]$  sería incorrectamente 1.

**Restricciones:** Ud. no puede usar los operadores de multiplicación, división o módulo (\* / %). Use eficientemente los operadores de bits. Está permitido calcular el resultado haciendo 8 sumas por medio de un ciclo de 8 iteraciones, pero se le descontará un punto. Hágalo astutamente con solo 2 sumas, recurriendo a un uso inteligente de máscaras 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 compara su eficiencia con la solución del profesor, (b) *sol-profesor* con el binario ejecutable de la solución eficiente del profesor, (c) *t1.h* que incluye el encabezado de la función pedida, y (d) *Makefile* que le servirá para compilar su tarea. Ud. debe programar la función *sumaVecs* en el

archivo *t1.c*. El archivo *t1.c.plantilla* muestra algunos *includes* que Ud. debe agregar en *t1.c*. Compile su tarea en alguna plataforma Linux (Ejemplo: Debian, Ubuntu, Ubuntu bajo WSL en Windows, Kali, etc.) con el comando *make* sin parámetros. Depure su tarea con el comando *ddd test-t1*. El programa de prueba lo felicitará cuando su solución sea correcta.

### Requisitos para aprobar la tarea

Ud. debe probar su tarea bajo *Debian 10* de 64 bits con los comandos *make* y *make test-O*. Este último usa las opciones de optimización del compilador de C. Su tarea será aprobada cuando el comando *make* termine mostrando el mensaje:

```
Felicitaciones: su solucion es correcta
```

Además el comando *make test-O* debe terminar mostrando el mensaje:

```
Felicitaciones: su solucion es correcta y eficiente
```

Esto indicaría que su solución no es 80% más lenta que la solución eficiente del profesor. Si este comando muestra el mensaje:

```
Su solucion es correcta, aunque no suficientemente
```

```
eficiente, sera aprobada pero se le descontara un punto
```

Su tarea será aprobada, pero se le descontará un punto como dice el texto porque supera en un 80% el tiempo de la solución eficiente. En todo caso su tarea será rechazada si toma 4 veces el tiempo de la solución eficiente o más. Si no aprueba la tarea, su nota es 1.0.

Observaciones:

- Está garantizado que el binario ejecutable *sol-profesor* corre en Debian 10, pero no necesariamente en otras distribuciones de Linux. Si *make test-O* falla en otras distribuciones de Linux, es normal. Esta prueba debe hacerla con Debian 10 de 64 bits.
- Su solución corre mucho más rápido con *make test-O* porque gcc fue invocado con la opción *-O3* para que optimice las instrucciones de máquina generadas.
- Su tarea será inspeccionada para revisar que cumple con las restricciones (no usar \* / %). Será rechazada si no las cumple.

### Entrega

Ud. solo debe entregar el archivo *t1.c* 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.