

## Parcialito Proyecto 1 - Tema B

Extender el esqueleto de código dado en el archivo `parcialito-skeleton.c` e implementar un programa que haga lo siguiente:

1. Lea un array de números naturales desde un archivo (dado en el esqueleto).
2. Lea un número no negativo desde la entrada estándar.
3. Imprima por pantalla la menor cantidad de elementos del array tal que su suma sea mayor o igual al número ingresado en el ítem anterior.

**Para el punto (1)**, que ya está resuelto en el esqueleto, asumir que el array descripto en el archivo pasado por parámetro contiene sólo naturales. Es decir, no hay que implementar ningún chequeo extra.

**Para el punto (2)**, utilizar `scanf` y manejar errores si el usuario no ingresó un número o si el número que ingresó es negativo. Es decir, si el usuario ingresa cualquier cosa que no sea cero o un número positivo, mostrar un mensaje de error por pantalla y terminar la ejecución con código de error.

El mensaje de error debe ser: `You should enter a non-negative number.`

**Para el punto (3)**, ordenar el array ascendentemente usando cualquiera de los algoritmos de ordenación ya implementados (usar el `sort.o` dado en el esqueleto). Luego, recorrer el array de ATRÁS para adelante e ir imprimiendo el elemento actual siempre que éste, sumado a lo acumulado hasta el momento, no supere o sea el primero tal que se supere al número ingresado en el ítem (2). Si la suma no alcanzara, reportar cuánto faltó.

### Compilación

Para usar los `array_helpers.o` y `sort.o` dados, usar la siguiente línea de compilación:

```
gcc -Wall -Werror -Wextra -pedantic -std=c99 array_helpers.o sort.o parcialito.c -o parcialito
```

### Ejemplo

Supongamos que el array pasado al programa es:

3 8 27 16 5 1

- Si el usuario ingresa por pantalla 0, el programa debería mostrar: 27
- Si el usuario ingresa por pantalla 2, el programa debería mostrar: 27
- Si el usuario ingresa por pantalla 30, el programa debería mostrar: 27 16
- Si el usuario ingresa por pantalla 43, el programa debería mostrar: 27 16
- Si el usuario ingresa por pantalla 44, el programa debería mostrar: 27 16 8
- Finalmente, si el usuario ingresa 100, como la suma de todos los elementos del array no supera este valor, el programa debería mostrar todos los elementos y el faltante (un número negativo, la diferencia entre la suma obtenida y el número ingresado): 27 16 8 5 3 1 -40

**OJO** Pensar en el array vacío también!

## Recordar

- Se debe resolver el ejercicio en una hora (o menos).
- Se deben usar las librerías dadas `array_helpers` y `sort`.
- Se debe compilar usando `gcc` desde la consola pasando todos los flags usados en el proyecto.
- La entrega es a través de Jaime usando la cuenta individual de cada alumno. El archivo a subir se debe llamar `parcialito.c`, y al subirlo no debería tener errores de compilación.
- Comentar e indentar el código apropiadamente, siguiendo el estilo de código ya provisto por la cátedra en los archivos dados (indentar con 4 espacios, no pasarse de las 80 columnas, inicializar todas las variables, etc).