

Supóngase que un producto comercial P está descrito por una secuencia de W bits que denotan la presencia o ausencia de características: si $P_i = 1$ entonces P cuenta con la i -ésima característica, mientras que si $P_i = 0$ entonces P no cuenta con ella. De forma similar, supóngase que un cliente desea encontrar un producto que sea compatible con su búsqueda B , la cual se describe por una secuencia de W bits donde $B_i = 1$ significa que el cliente desea un producto que tenga la i -ésima característica, mientras que $B_i = 0$ significa que la ausencia o presencia de dicha característica en un producto es irrelevante para la búsqueda.

Escribe un función `cuenta_exitosas` que reciba un arreglo de N productos y un arreglo de M búsquedas y que regrese un entero que denote cuántas de las búsquedas lograron encontrar al menos un producto compatible. Tanto los productos como las búsquedas se representarán como valores de tipo `int` vistos como arreglos de bits (por lo que puedes suponer que $W = 32$). Los arreglos se representarán usando `std::vector`. Puedes suponer que $0 \leq N \leq 2^{19}$, que $0 \leq M \leq 2^{18}$ y que tanto N como M son potencias de 2.

Un ejemplo de función secuencial que resuelve el problema está disponible en <https://racc.mx/uam/home/2022-o/tslp/secuencial.cpp>. Si tu función no responde correctamente todos los casos de evaluación, entonces tu calificación será 0. Si tu función responde correctamente todos los casos de evaluación, entonces tu calificación se calculará como sigue: 0 puntos si tu función es igual o más lenta que la función de ejemplo y 100 si tu programa es al menos cuatro veces más rápida que la función de ejemplo cuando tu función se ejecuta en una computadora con **ocho núcleos físicos**; los tiempos de ejecución intermedios recibirán un puntaje proporcional.

Tu código no declarar `main` y no debe usar variables estáticas o globales. Se permite incluir cualquier archivo de la biblioteca de C++ y se permite declarar tipos y funciones auxiliares, así como constantes globales en tiempo de compilación. Cada función puede declarar variables auxiliares que no superen los 500 kilobytes en total por función. Puedes consultar una programa secuencial de prueba en https://racc.mx/uam/home/2022-o/tslp/secuencial_prueba.cpp y una instancia de ejemplo en https://racc.mx/uam/home/2022-o/tslp/ejemplo_instancia.txt. Deberás enviar el código fuente de tu tarea (función `cuenta_exitosas`, definiciones auxiliares e inclusiones de biblioteca) desde tu cuenta institucional al formulario <https://forms.gle/gyEwgJeb58V1KnK4A>.

Ejemplo de uso	Ejemplo de salida
<pre>std::vector<int> productos = { 8, 4, 9, 2, 7, 3 }; std::vector<int> busquedas = { 5, 0, 15, 6 }; std::cout << cuenta_exitosas(productos, busquedas);</pre>	3