Inciso c

Resultado de las validaciones de las matrices en ambos lenguajes:

```
alulab@V20805:~/Documentos/E1_20212126$ gcc diag_asm.o preg_diag.c -o main
alulab@V20805:~/Documentos/E1_20212126$ ./main
Matriz 1:
------------

[  1, 2, 3, ],
[  4, 5, 6, ],
[  7, 8, 9, ],

Resultado en C es: 1
Resultado en Asm es: 1
Matriz 2:
------------

[  0, 1, 0, ],
[  1, 0, 1, ],
[  0, 1, 0, ],

Resultado en C es: 0
Resultado en Asm es: 0
```

Inciso d

A partir de los benchmarks y de considerar como tiempo referencial a $10^4 n$. Se tienen los siguientes ratios y speedup:

```
Benchmarks:
-----------------

Tiempo de ejecución N°1 en C es 80486.000 ns
Tiempo de ejecución N°1 en Asm es 30339.000 ns

Tiempo de ejecución N°2 en C es 9227.000 ns
Tiempo de ejecución N°2 en Asm es 10087.000 ns

Tiempo de ejecución N°3 en C es 21286.000 ns
Tiempo de ejecución N°3 en Asm es 5533.000 ns

Tiempo de ejecución N°4 en C es 11759.000 ns
Tiempo de ejecución N°4 en Asm es 6060.000 ns

Tiempo de ejecución N°5 en C es 20084.000 ns
Tiempo de ejecución N°5 en Asm es 5083.000 ns



Ratio C: 2.063
Ratio en Asm: 0.878
SpeeUp (C/Asm): 2.349
```

Inciso e

Del inciso “d” se evidencia que el código en ensamblador es más eficiente que en C a partir de la SpeedUp. Ambos fueron hechos casi por el mismo algoritmo; sin embargo, posiblemente las diferencias se deban a la manera que se direccionaron los datos, en ensamblador se utilizó 3 veces un mismo resultado de multiplicación necesario para el offset. En C no se hizo de la misma manera y por lo tanto, tiene que multiplicar los mismos resultados, generando gastos de tiempo innecesarios.