

1. Dado el siguiente código:

```
#define N 512
float a[N][N], b[N], t=0;
int i, j;
int main() {
    for(i=0; i<N ; i++){
        t+= b[i];
        for (j=0; j<N; j++)
            t+= a[j][i];
    }
}
```

a) [0,125p] ¿Es legal aplicar la técnica de intercambio de bucles? ¿por qué?

b) [0,125p] ¿Y es beneficioso? ¿por qué?

c) [0,5p] Compila con -O2 dos versiones de l código, una sin modificación alguna y la otra aplicando el intercambio de bucles. Muestra en dos columnas el número de fallos que se producen en ambas versiones del código las referencias a los dos vectores, a y b, en la caché de datos de nivel 1 (1024 bytes y correspondencia directa) a medida que aumentamos el tamaño de línea, comenzando con 32 bytes.

Versión antes de aplicar intercambio de bucles

Tamaño de línea	Fallos a	Fallos b
32		
64		
128		

Versión después de aplicar intercambio de bucles

Tamaño de línea	Fallos a	Fallos b
32		
64		
128		

2. [0.25p] Dado el siguiente código

```
#define N 256
double a[N][N], t;
int i, j;
int main() {
    for(i=0; i<N ; i++)
        for (j=0; j<N; j++)
            t=t*0.9+ a[i][j];
}
```

Considerando una caché de correspondencia directa de 4096 bytes y líneas de 32 bytes, ¿cuál es el mínimo relleno a aplicar para que se pueda beneficiar de la técnica de relleno de arrays? Justificalo.

Solución:

1.

Nota: Para compilar

`gcc -g -O2 nombre_fichero.c`

`valgrind -v --D1=TamañoCache,Asociatividad,TamañoLinea --tool=cachegrind ./a.out`

para ver los fallos de cada instrucción con `cg_annotate --auto=yes ./cachegrind.out.numeroCodigo` siendo numero codigo el número que aparece a la izquierda en todas las líneas de valgrind

a) Es legal . No varía el resultado calculado.

b) Es beneficioso. Se haría un acceso por filas en la matriz y como las matrices se almacenan por filas en C, se producirían menos fallos de caché.

c)

a tiene 512×512 elementos

Versión antes de aplicar intercambio de bucles

Tamaño de línea	Fallos a	Fallos b
32	262144	64
64	262144	257 ó 288
128	262144	384

Versión después de aplicar intercambio de bucles

```
#define N 512
float a[N][N], b[N], t=0;
int i, j;
int main() {
    for(j=0; j<N ; j++){
        t+= b[j];
        for (i=0; i<N; i++)
            t+= a[j][i];
    }
}
```

Tamaño de línea	Fallos a	Fallos b
32	32768	512
64	16384	512
128	8256	512

2.

Cada fila ocupa $256 \times 8 = 2048$ bytes $\Rightarrow 4096 / 2048 = 2$ accesos

Cada 2 accesos van a parar a la misma línea de caché.

Como cada línea alberga 4 elementos de la matriz, y cada 2 accesos debemos desplazarnos una línea $\Rightarrow 4 / 2 = 2$ elementos de relleno en cada fila.