

Donada la següent funció `foo` en llenguatge C:

```
void foo(short M[][64], unsigned int k) {
    int i;
    short aux = M[k][8];
    for (i=5; i<45; i+=5){
        M[45-i][i] = aux;
    }
}
```

Completa el següent codi MIPS omplint les caselles en blanc perquè sigui equivalent a l'anterior codi en alt nivell, tenint en compte que els elements de la matriu `M` s'accedeixen utilitzant la tècnica d'accés seqüencial sempre que es pot, usant el registre `$t1` com a punter. Aquest punter `$t1` s'inicialitza amb l'adreça de l'element `M[40][5]`. Al codi se li ha aplicat l'optimització de conversió d'un bucle `for` en un `do_while` i l'eliminació de la variable d'inducció.

```
    sll    $t0, $a1, 
    addu   $t0, $t0, $a0
    lh     $t2, ($t0)           # aux = M[k][8];
    addiu  $t1, $a0,            # @M[40][5]
    addiu  $t3, $a0,            # adreça final del punter $t1
    b      cond
bucle:   sh     $t2, ($t1)
    addiu  $t1, $t1, 
cond:    bgtu   $t1, $t3, bucle
    jr     $ra
```