

Problem 49. (10 points):

After a stressful semester you suddenly realize that you haven't bought a single christmas present yet. Fortunately, you see that one of the big electronic stores has CD's on sale. You don't have much time to decide which CD will make the best presents for which friend, so you decide to automatize the decision process. For that, you use a database containing an entry for each of your friends. It is implemented as an 8×8 matrix using a data structure `person`. You add to this data structure a field for each CD that you consider:

```
struct person{

    char name[16];

    int age;
    int male;

    short nsync;
    short britney_spears;
    short dolly_parton;
    short garth_brooks;
}

struct person db[8][8];
register int i, j;
```

Part 1

After thinking for a while you come up with the following smart routine that finds the ideal present for everyone.

```
void generate_presents(){
    for (j=0; j<8; j++){
        for (i=0; i<8; i++) {
            db[i][j].nsync=0;
            db[i][j].britney_spears=0;
            db[i][j].garth_brooks=0;
            db[i][j].dolly_parton=0;
        }
    }

    for (j=0; j<8; j++){
        for (i=0; i<8; i++) {

            if(db[i][j].age < 30){
                if(db[i][j].male)
                    db[i][j].britney_spears = 1;
                else db[i][j].nsync = 1;
            }
            else{
                if(db[i][j].male)
                    db[i][j].dolly_parton = 1;
                else db[i][j].garth_brooks = 1;
            }
        }
    }
}
```

Of course, runtime is important in this time-critical application, so you decide to analyze the cache performance of your routine. You assume that

- your machine has a 512-byte direct-mapped data cache with 64 byte blocks.
- db begins at memory address 0
- The cache is initially empty.
- The only memory accesses are to the entries of the array db. Variables i, and j are stored in registers.

Answer the following questions:

A. What is the total number of read and write accesses? _____.

B. What is the total number of read and write accesses that miss in the cache? _____ .

C. So the fraction of all accesses that miss in the cache is: _____.

Part 2

Then you consider the following alternative implementation of the same algorithm:

```
void generate_presents(){
    for (i=0; i<8; i++){
        for (j=0; j<8; j++) {
            if(db[i][j].age < 30)
                if(db[i][j].male) {
                    db[i][j].nsync=0;
                    db[i][j].britney_spears=1;
                    db[i][j].garth_brooks=0;
                    db[i][j].dolly_parton=0;
                }
            else
                db[i][j].nsync=1;
                db[i][j].britney_spears=0;
                db[i][j].garth_brooks=0;
                db[i][j].dolly_parton=0;
        }
    else{
        if(db[i][j].male) {
            db[i][j].nsync=0;
            db[i][j].britney_spears=0;
            db[i][j].garth_brooks=0;
            db[i][j].dolly_parton=1;
        }
        else{
            db[i][j].nsync=0;
            db[i][j].britney_spears=0;
            db[i][j].garth_brooks=1;
            db[i][j].dolly_parton=0;
        }
    }
        }
    }
```

Making the same assumptions as in Part 1, answer the following questions.

- A. What is the total number of read and write accesses? _____
- B. What is the total number of read and write accesses that miss in the cache? _____
- C. So the fraction of all accesses that miss in the cache is: _____.