PrairieLearn CS 233, Sp25 Che Liu ▼ Assessments Gradebook PQ6#1

## Question 8: Cache Performance Analysis

For a **32**KB **1**-way set associative cache with **64**-byte blocks on a machine with **32**-bit address space, consider the following code snippet. The cache uses a Least Recently Used replacement policy. Note: The memory is byte addressable as always and data is stored in row-major order. Assume that the cache is initially empty.

You may also assume that expressions in C code get evaluated in the order from left to right - e.g. a += b \* c accesses a then b then c then a.

double and long long are 8B; float and int are 4B; short are 2B, and char are 1B.

```
#define N 151
struct pokemon {double level; char hp;};
struct pokemon all_pokemon[N];
int hp_needed = get_needed_hp();
for (int i = 0; i < N; i++) {
    if (all_pokemon[i].level < 100) { // IF 1</pre>
        if (all_pokemon[i].hp >= hp_needed) { // IF 2
            all_pokemon[i].level ++;
```

Only data structures all\_pokemon is stored in memory, and all other variables are register allocated. all\_pokemon is stored starting at address 0x0A390848.

Part(a) Assuming that IF 1 is true for the first 80 times and IF2 is true for the first 17 times, compute the number of accesses and misses for all\_pokemon.

Number of Accesses	Number of Misses
all_pokemon:	all_pokemon:

Part(b) How would the number of misses change if:

(i) level was of type short.

Number of Misse	S	
all_pokemon:		8

(ii) hp was of type float.

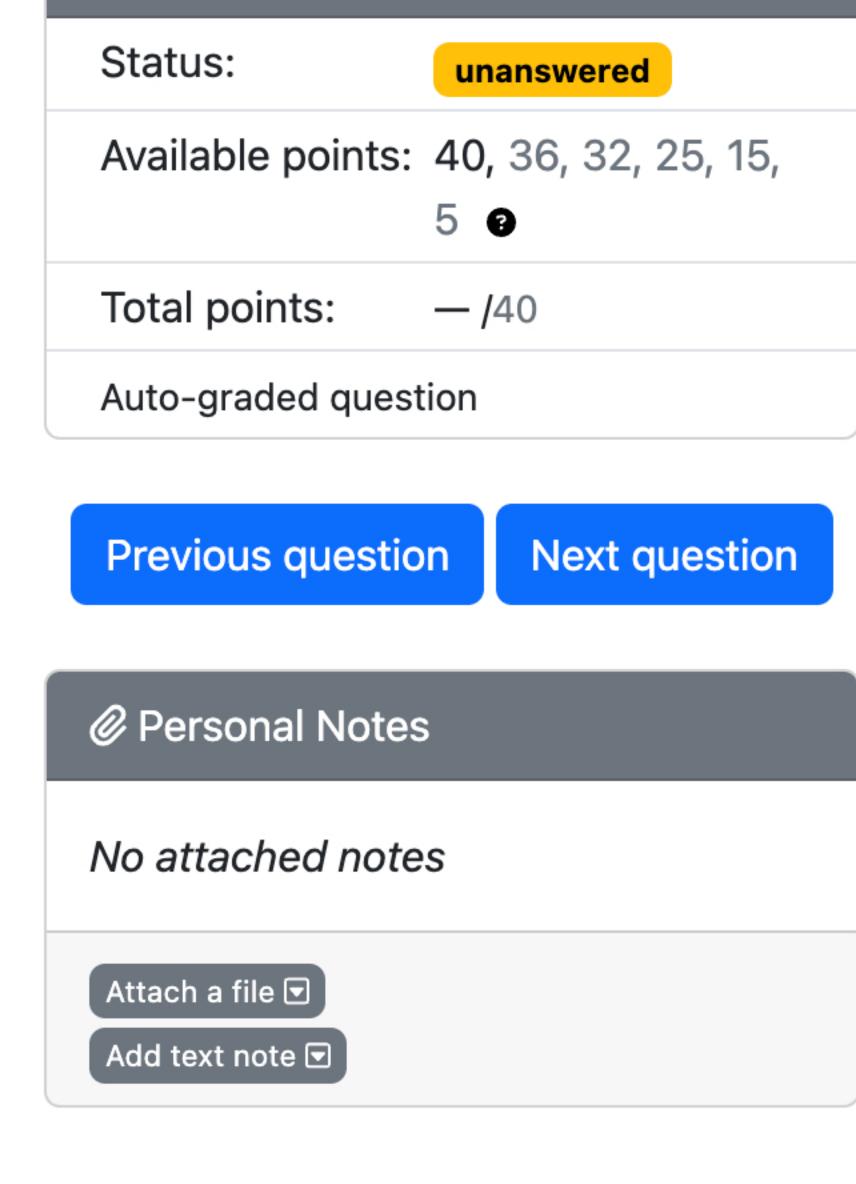
Save & Grade 6 attempts left

N	lumber of Misse	S	
	all_pokemon:		

Part(c) For the original code, consider now how the behavior of the code changes if the compiler didn't register allocate hp\_needed, and instead left it in memory at address 0x3B260848 and it has to be loaded each iteration of the loop. Compute the number of misses, assuming both the IF statements are always true.

Save only

|--|



Assessment overview

Practice Quiz 6

Question 8







HEX

DEC

COPY