Page 1 of 0

Problem 18. (14 points):

Consider the source code below, used to keep track of the rooms currently reserved in a family-run hotel. Each entry in the `residents` array stores a name of the customer reserving the room. `FLOORS` represents the number of floors in the hotel. `ROOMS` represents the number of rooms per floor. Both are constants declared with `#define`. LEN, the maximum number of bytes allocated for a name, is defined to be 12.

```
char residents[FLOORS][ROOMS][LEN];

void
reserve_room(int floor, int room, char *custname)
{
        strcpy(residents[floor][room], custname);
}
```

The assembly code for the function `reserve_room` looks like this:

```
reserve_room:
        pushl %ebp
        movl %esp,%ebp
        movl 12(%ebp),%eax
        movl 16(%ebp),%edx
        pushl %edx
        movl 8(%ebp),%edx
        sall $4,%edx
        subl 8(%ebp),%edx
        leal (%eax,%eax,2),%eax
        leal residents(,%eax,4),%eax
        leal (%eax,%edx,4),%edx
        pushl %edx
        call strcpy
        movl %ebp,%esp
        popl %ebp
        ret
```

A. What is the value of `ROOMS`?

B. Due to a strange bug, the program accesses `residents[0][1][-2]`. What value is actually being accessed? (Express your answer as an *integer triplet (-,-,-)*. You may assume that FLOORS and ROOMS are both greater than 1)

C. The programmer realizes that this implementation is wasteful of memory. Successive fires in several memory chip factories in Taiwan drive up memory prices and finally convince him to improve the memory efficiency of his implementation to maintain the competitiveness of the family hotel.

The declaration of `residents` is changed to be a two dimensional array of pointers to character strings (names). The new code allocates memory for customer names only for those rooms that are actually reserved. Otherwise, `residents[f][r]` stores a NULL pointer. **For simplicity, assume there is no storage overhead due to `malloc`.**

The new declaration looks like this:

```
char *residents[FLOORS][ROOMS];

void
reserve_room(int floor, int room, char *custname)
{
    residents[floor][room] = malloc(LEN);
    strcpy(residents[floor][room], custname);
}
```

After a few months. The programmer goes back to review the memory savings of his improved scheme. During that period, the hotel was 20% reserved. The programmer is delighted because the savings are found to be 168 bytes! How many floors does this hotel have? (that is, what is the value of FLOORS?)