STEFAN MALANIK
12.01.2021.

DSA Exam S2

1. // struct for flights -> implemented as an unordered list

```
typedef struct flight {
    char name[30];
    struct flight *next; // linked-list
    struct pass* head; // -> this struct should be the first in code to have the implementation for passengers
} flight;
// struct for passengers -> implemented as an ordered list
typedef struct pass {
    long code; // enough for 8 digits
    struct pass *next; // linked-list
} pass;
flight * list; // main list as global pointer
void remove_pass ( long code) {
    flight *p = list; // go through the flight list
    while (p != NULL) {
        // check for each flight for the specific passenger
        pass* pass_list = p -> head;
        while( pass_list != NULL){ // iterate through each passenger
            if(pass_list -> code == code) {
```

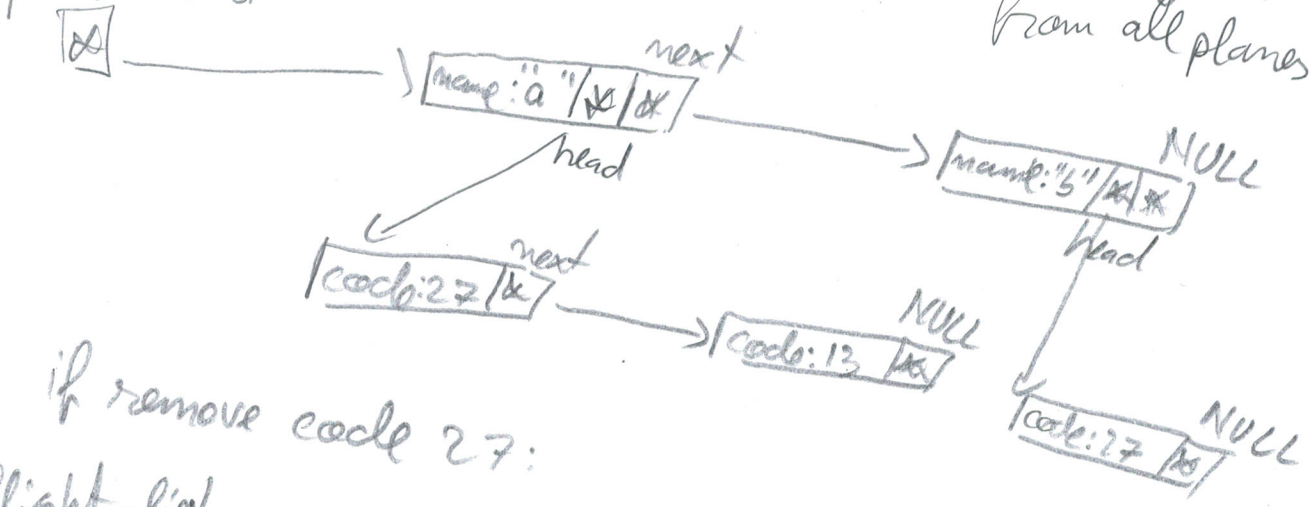
```
// remove pass_list entry
pass* next = pass_list->next; // if last elem, this will be null
// check if pass_list is head of list
if (pass_list == p->head) {
    p->head = next;
}
// free memory and create a link
```

```
// iterate with the next pointer to keep last occurence
// in memory
// first pointer -> edge case
if (pass_list != NULL && pass_list->code == code) {
    pass* next = pass_list->next;
    free(pass_list);
    p->head = next;
}
// now iterate with next pointer if pass_list not null
if (pass_list == NULL) continue;
while (pass_list->next != NULL) {
    // if condition for removal met on next, remove it.
    if (pass_list->next->code == code) {
        pass* nextp = pass_list->next->next;
        free(pass_list->next);
        pass_list->next = next; // link the list back
    }
    pass_list = pass_list->next;
}
```

1.

```
        // after this while, we go to the next flight in the list
        p = p->next;
        }
    }
```

this algorithm has $O(M)$ complexity where M is total number of passengers from all planes.



if remove code 27: