TATU BOGDAN
19.02.2021

```
class Conference {
    unsigned id;
    string password;
    Student * head;
    Conference * next;
}
class Student {
    string name;
    Student * next;
}

void delete_students (Conference * head, char n) {
    Conference * node = head;
    for (; node != NULL; node = node->next)
    { // foreach conference
        Student * prev, * curr = node->head;
        for (; curr != NULL; curr = curr->next)
        { // finds student with name ending in n
            if ( curr->name [name.len-1] == n )
            {
                if (curr == node->head)
                { // case if curr is head
                    node->head = node->head->next
                    free (curr);
                }
            }
        }
    }
}
```

```
        else { // case if curr isn't head
            prev -> next = curr -> next;
            free (curr);
        }

    }

    prev = curr;

}

main () {
    Conference *head = get-conference ();
    cin >> n;
    delete - students (head, n);

}
```

Complexity: $O(n*m)$

Data Structs: Conference & 8 Student - Lists
    because we don't have a fixed nr of confs/
    students

Ex: C1 — C2 — C3 — C4   if n = 1
       ~~S1~~   S2   S4   S3
              ~~S4~~   S5   S3
                    S6   ~~S7~~