

Programmieren 1

Auditorium Exercise 10

Vorlesungsfreie Zeit 21.12.20 bis 02.01.21

- Präsenzübungen finden in diesem Jahr bis einschließlich den 19.12.20 statt. Im nächsten Jahr beginnen Sie ab dem 04.01.21 wieder.
- Besprechung von Assignment 09 in der Woche vom 04.01.21 – 08.01.21
- Abgabe von Assignment 10 bis zum 07.01.21 um 23:59 Uhr
- Nächster Live Termin am 08.01.21 wie gewohnt ab 9 Uhr

LernLounge

- Studentische Tutorinnen und Tutoren helfen Ihnen und unterstützen Sie beim Lernen
- fachliche Fragen, Fragen zum selbstorganisierten Lernen
- Online: <http://go.lu-h.de/jHcTY>
- Fächer-Schwerpunkte
 - Programmieren
 - Analysis, Lineare Algebra
 - Grundlagenfächer 1. und 2. Semester

Koordination: Isabelle Kross, Tel.: 0511 762 3744, Email: isabelle.kross@et-inf.uni-hannover.de

LernLounge (ab 26.10.2020)

Zeit		Dienstag	Mittwoch	Donnerstag
10:00 – 10:30		Alexander	Malte	
10:30 - 11:00		Alexander	Malte	
11:00 – 11:30				
11:30 – 12:00				
12:00 – 12:30				
12:30 – 13:00				
13:00 – 13:30			Lennart	
13:30 – 14:00			Lennart	
14:00 – 14:30	Dana		Dana	Lennart
14:30 – 15:00	Dana		Dana	Lennart
15:00 – 15:30	Dana		Dana	Lennart
15:30 – 16:00				Lennart

Schwerpunktfächer

Alexander

- Programmieren
- GTI

Dana (ab 02.11.2020)

- Lineare Algebra
- Analysis

Malte

- GTI
- GDS

Lennart

- Programmieren
- Lineare Algebra

Last Minute Help for your Submission

- Who?
 - Maximilian Lumpe & Lukas Köhler
- When?
 - On thursdays from 6pm to 8pm (Maximilian) & 6 pm to 7pm (Lukas)
- Where?
 - On our Discord Server
- How?
 - Enqueue into the queue of him and wait your turn (like enqueueing for your TA slot)

Questions?

Assignment 9 – Task 3a

```
int length_list(Node* list) {  
    int i = 0;  
    for (Node* n = list; n != NULL; n = n->next)  
        i++;  
    return i;  
}
```

Assignment 9 – Task 3b

```
String get_list(Node* list, int position) {
    if (position < 0)
        return NULL;
    Node* n = list;
    int i = 0;
    for (; n != NULL; n = n->next, i++) {
        if (i == position)
            return n->value;
    }
    return NULL;
}
```


Assignment 9 – Task 3c

```
void free_list(Node* list) {
    Node* node_next = NULL;
    for (Node* node = list; node != NULL; node = node_next) {
        node_next = node->next;
        if (node->value != NULL) {
            free(node->value);
        }
        free(node);
    }
}
```

Assignment 9 – Task 3d

```
Node* append_list(Node* list, String value) {
    if (list == NULL) { // empty list
        return new_node(value, NULL);
    } else { // non-empty list
        Node* n = list;
        while (n->next != NULL)
            n = n->next; // find last element
        n->next = new_node(value, NULL);
        return list;
    }
}
```

Assignment 9 – Task 3e

```
void print_situation(void)
{
    prints("fertige Essen: ");
    println_list(food);
    printf("nächster Essenswunsch: %s", students->value);
    printf(" (%d hungrige Studierende warten)\n", length_list(students));
    printf("Reputation der Mensa: %d\n> ", reputation);
}
```

Assignment 9 – Task 3f

```
void finish(void) {  
    println("Fertig für heute. Die Mensa schließt.");  
    printf("Finale Reputation der Mensa: %d\n", reputation);  
    free_list(food);  
    free_list(students);  
    exit(0);  
}
```

Assignment 9 – Task 3g

```

void run_mensa(void) {
    // create 5 random food items from the menu
    food = NULL;
    for (int i = 0; i < 5; i++) {
        food = new_node(menu[i_rnd(menu_count)], food);
    } // create 3 random food wishes from the menu (each wish from one
student)
    students = NULL;
    for (int i = 0; i < 3; i++) {
        students = new_node(menu[i_rnd(menu_count)], students);
    }
    print_situation();

    //...

```

Assignment 9 – Task 3g

```
void run_mensa(void) {//...
    int i;
    while ((i = i_input()) >= -1) {
        String s = students->value;
        if (i == -1) {
            printf("%s ist nicht da? Schade.\n", s);
            reputation -= 2;
        } else {//...

//...
    }
```

Assignment 9 – Task 3g

```
void run_mensa(void) {//...
    //...
    } else {
        String f = get_list(food, i);
        if (s_equals(f, s)) {
            println("Vielen Dank! Ich liebe die Mensa!");
            food = remove_list(food, i);
            reputation += 1;
            students = append_list(students,
                                   menu[i_rnd(menu_count)]);
            food = append_list(food, menu[i_rnd(menu_count)]);
        } else {
            //...
        }
    }
}
```

Assignment 9 – Task 3g

```

void run_mensa(void) {//...
    //...
    } else {
        String f = get_list(food, i);
        if (s_equals(f, s)) {//...
        } else {
            printf("%s möchte ich nicht! Ich möchte %s!\n", f, s);
            reputation -= 1;
        }
    }
    students = remove_list(students, 0);
//...
}

```


Assignment 9 – Task 3g

```
void run_mensa(void) {  
    //...  
    students = remove_list(students, 0);  
    if (length_list(students) <= 0)  
        break;  
    print_situation();  
}  
finish();  
}
```

Assignment 10

- Will be available on StudIP this afternoon
- We will have a brief look inside now

Good luck and have fun!



programmieren1@hci.uni-hannover.de