

## Hand-in week 6

Github: <https://github.com/Dan-jpg2/assignment-6-Dan-jpg2-main>

- (1) (Text answer) (old exam question) Consider the following program fragment:

```
1  int x;  
2  int y;  
3  int z;  
4  int* w;  
5  int* q;  
6  x = 0;  
7  y = 1;  
8  z = 2;  
9  w = &x;  
10 q = &y;  
11 *w = y;  
12 *q = z;  
13 *w = x + y + z + *q;  
14 *q = x + y + z + *w;  
15 printf("x=%d, y=%d, z=%d", x, y, z);
```

What does the program print when it is executed?

Answer:

$x = 7, y = 18, z = 2$

- (2) (code answer) Write a function `int max(int* numbers, int size)` that, given an array of numbers (and its size), finds the maximum value in the array. You may

1

assume that the array is not empty (i.e. `size > 0`). Include assertions in the implementation of `max` to ensure that the precondition is fulfilled when executing the function

See code in link to github.

(3) (Text and Code answer) Consider the following program:

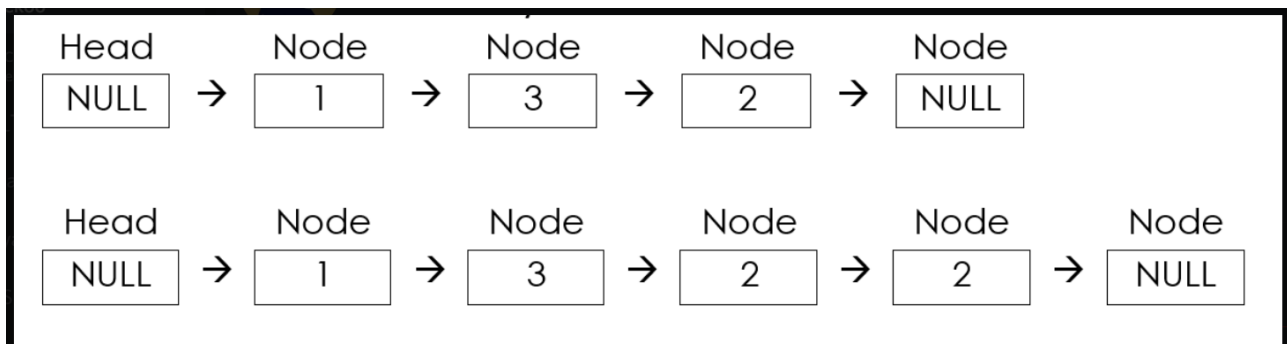
```

1 #include <stdio.h> /*printf*/
2 #include <assert.h> /*assert*/
3 #include <stdlib.h> /*malloc*/
4
5 typedef struct node {
6     int data;
7     struct node *next;
8 } node;
9
10 void add(node *head, int x){
11     /*pre: head points to the first, empty element.
12        The last element's next is NULL
13        post: a new node containing x is added to the end of the list*/
14     assert(head!=NULL);
15     node *p = head;
16     while (p->next!=NULL) {
17         p = p->next;
18     } /*p points to the last element*/
19     node *element = malloc(sizeof(node));
20     element->next = NULL;
21     element->data = x;
22     p->next = element;
23 }
24
25 int main(void) {
26     node *list = malloc(sizeof(node));
27     list->next = NULL; /*create first, empty element*/
28     add(list,1);
29     add(list,3);
30     add(list,2);
31     /*show list here*/
32     add(list,2);
33     /*show list here*/
34     return 0;
35 }

```

- (a) Draw two diagrams that shows list at /\*show list here\*/ in main.  
**Note:** The first element is empty and holds no data. I.e. if I have a list with two elements, it has three nodes (the first, empty one and then two nodes holding data). The same definition is used in all functions.

Below is the diagram of the two instances in task a)



- (b) Implement a function with the following signature: `int size(node *l)`. It has the same precondition as `add` and returns the number of elements in the list. E.g. if `size(list)` was printed out at the first `/*show list here*/` in main, the result would be 3.

See code on github.

- (c) What does the following code do when executed? (i.e. do the code fulfil the post condition? If not, what happens?)

```
void printout(node *l) {
    /*pre: head points to the first, empty element.
       The last element's next is NULL
    post: the values of the list are printed out*/
    node *p = l->next;
    while (p!=NULL){
        printf("%d, ",p->data);
    }
    printf("\n");
}
```

Koden opfylder ikke "post condition" da `p->data` ikke er blevet opdateret / fået en ny værdi. Det gør jeg ved at implementere "`p=p->next;`" se eventuelt koden for svar.

- (d) Correct the function above so that the post condition is fulfilled

See code on github

- (e) Write a function `int largest(node *l)`. The pre- and post conditions are the following:

```
/*pre: head points to the first, empty element.
   The last element's next is NULL. size(l>0)
post: returns the largest value of the list*/
```

See code on github