

(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:

1. Her får vi givet $x = 0$, $y = 1$, $z = 2$, $w = \&x$ (0) og $q = \&y$ (1).

$*w = y = 1$ dette ændrer også x til 1.

$*q = z = 2$ dette ændrer også y til 2.

$*w = x(1) + y(2) + z(2) + *q(2) = 7$ x bliver også ændret til 7.

$*q = x(7) + y(2) + z(2) + *w(7) = 18$ y bliver også ændret til 18.

Så svaret bliver: $x = 7$, $y = 18$, $z = 2$

(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

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

2. Pushed to github.

(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.

(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.

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

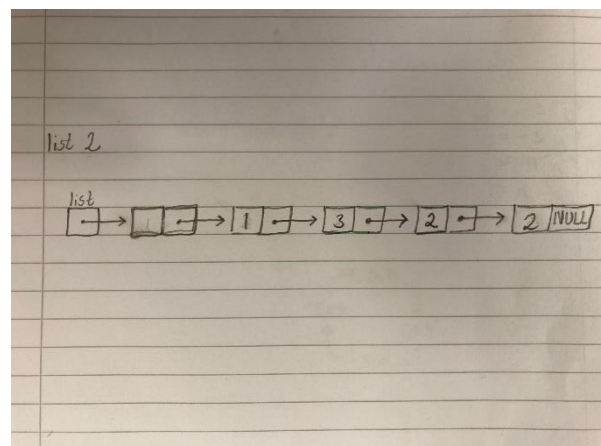
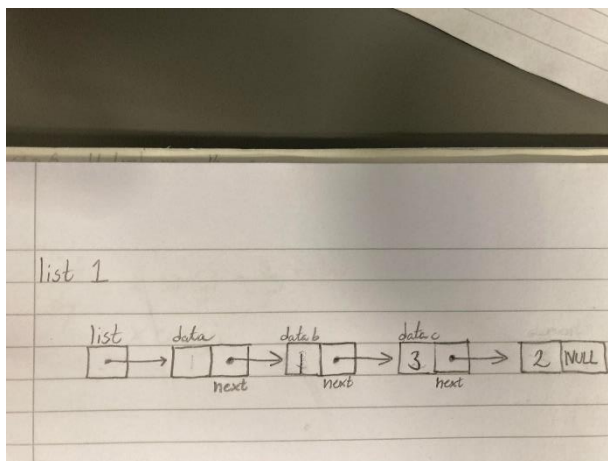
```
void printout (node *l) {  
    /*pre: head points to the first, empty element.  
       The last element's next is NULL  
    post: the values of the list are printed out*/  
    node *p = l->next;  
    while (p!=NULL){  
        printf("%d, ",p->data);  
    }  
    printf("\n");  
}
```

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

3.

a. Diagrams



- b. Check github.
c. Den looper uendeligt da p's værdi ikke bliver opdateret.
d. Check github.
e. Check github