

# Programming for Computerteknologi

## Hand-in Assignment Exercises

Week 6: Programming with pointers

1. (Text answer) 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:

The program prints the following, here are the calculations behind the program.

$$\begin{aligned}x &= 0, y = 1, z = 2 \\*w &= 1 \\*q &= 2 \\*w &= 1 + 2 + 2 + 2 \\*q &= 7 + 2 + 2 + 7\end{aligned}$$

The final print is then equal to:

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

2. (Code answer) Write a function `intmax(int *numbers, int size)`

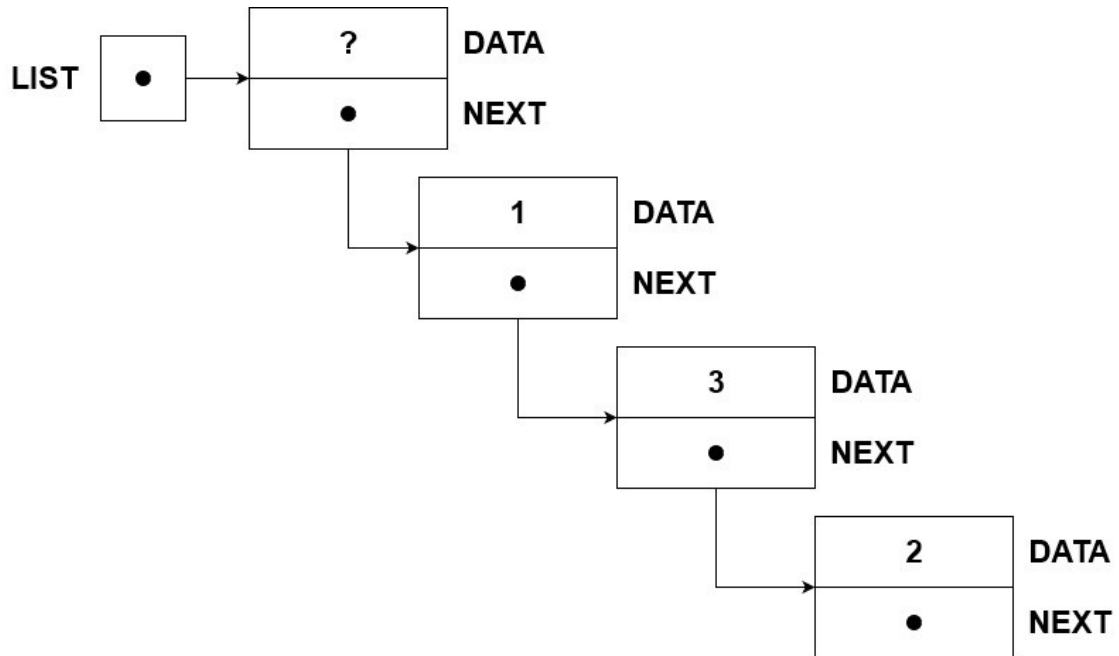
That given an array of numbers and its size, finds the maximum value in the array. You may assume that the array is not empty, (i.e., size > 0). Include assertions in the implementation of max to ensure that the precondition is fulfilled when executing the function.

3. (Text and code answer) Consider the following program:

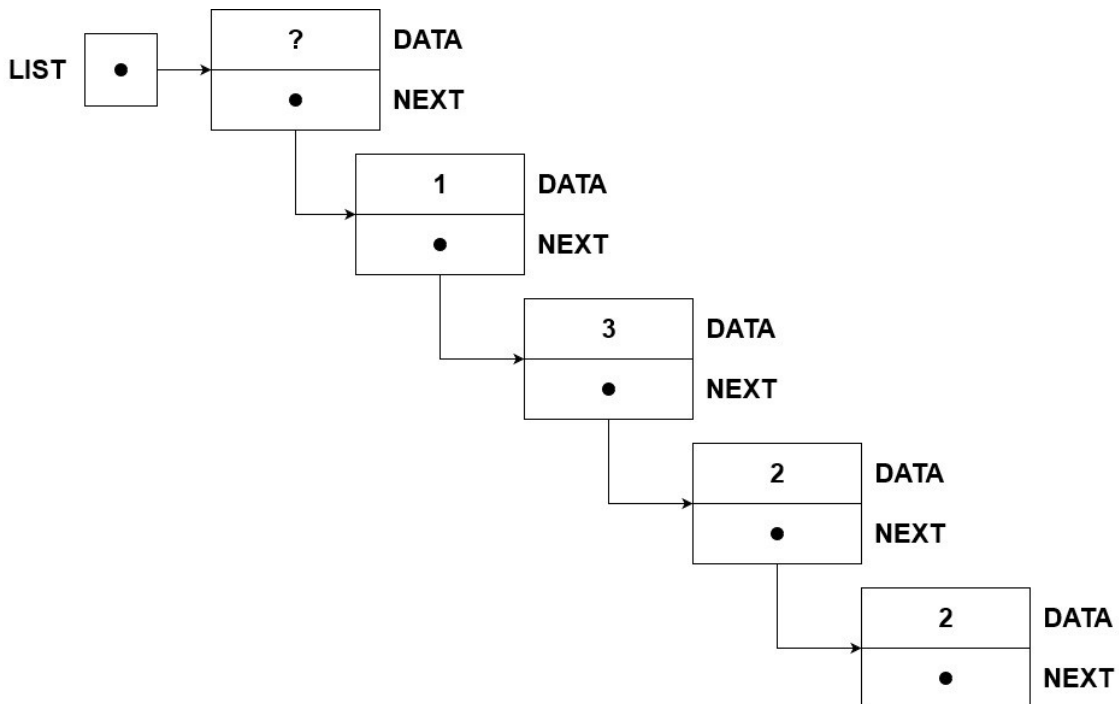
```
1 #include <stdio.h> /*printf*/
2 #include <assert.h> /*assert*/
3 #include <stdlib.h> /*malloc*/
4
5 typedef struct node {
6     int data;
7     struct node *next;
8 } node;
9
10 void add(node *head, int x){
11     /*pre: head points to the first, empty element.
12         The last element's next 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 show *list*

First diagram



Second diagram



- b) Implement a function with the following signature:

See the photo or the code in github.

```
int size(node *l){
    int num = 0;
    node *p = l->next;
    while(p->next != NULL){
        num++;
        p = p->next;
    }
    return num;
}
```

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

It runs an infinite loop since the pointer  $p$  is not redefined. Therefore, it would print the value in  $p \rightarrow data$  over and over. To fix this it would be necessary to redefine  $p$  in every loop to the next value.

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

I made sure  $p$  is redefined in every loop to ensure the correct  $p \rightarrow data$  while the loop is ran next.

```
p = p->next; /* My changes */
```

- e) Write a function *int largest(node \* l)*.

See the function in list.c,