

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  
7  x = 0;  
8  y = 1;  
9  z = 2;  
10 w = &x;  
11 q = &y;  
12 *w = y;  
13 *q = z;  
14 *w = x + y + z + *q;  
15 *q = x + y + z + *w;  
16  
17 printf("x=%d, y=%d, z=%d\n", x, y, z);
```

? Question

What does the program print when it is executed?

x= 0

y= 1

z= 2

w = &x

q = &y

*w=x=y=1 x=1

*q=y=z=2 y=2

*w=x =1+2+2+2 = 7 x=7

*q = 7+2+2+7 = y=18

z = 2

Programmet vil printe x=7, y=18, z=2

2) Se vscode

3)

a)

```
28 int main() {
29     node *list = malloc(sizeof(node));
30     list->next = NULL; // create first empty element
31     add(list, 1);
32     add(list, 3);
33     add(list, 2);
34     // Show list here
35     add(list, 2);
36     // Show list here
```

Da det første element i listen er tomt vil den liste som bliver lavet ved linje ved linje 33 være
[, 1, 3, 2]

og ved linje 35

[, 1, 3, 2, 2]

b)

```
// exercise 3.b
int size(node *l) {
    //Pre: l points to first empty element.

    //Post: size of the list
    assert(l!=NULL);

    int count = 0; //Initialize a counter variable as 0
    node *p = l->next; //Create a pointer to the "next" component of the node l
    while (p!=NULL){ //While loop incrementing count for every list component that is not NULL
        count++;
        p = p->next; //Update the pointer
    }

    return count;
}
```

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

```
1 void printout(node *l) {
2     // pre: head points to the first, empty element.
3     //       The last element's next is NULL
4     // post: The values of the list are printed out
5     node *p = l->next;
6     while (p != NULL) {
7         printf("%d, ", p->data);
8     }
9     printf("\n");
10 }
```

Det er meningen at koden printer alle data komponenterne for hvert element i listen. Men eftersom at koden ikke opdaterer pointeren er while loopet et uendeligt loop som søger efter en ikke eksisterende dataværdi at printe (det første element er tomt, så der er intet at printe).

Altså er postbetingelsen ikke opfyldt.

d)

```
// exercise 3.c and 3.d
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);
        p = p->next;
    }
    printf("\n");
}
```

I stedet skal pointeren opdateres sådan at den peger på det næste element i listen som set i denne kode.

e)

```
// exercise 3.e
int largest(node *l) {
    // pre: head points to the first, empty element.
    //       The last element's next is NULL.
    // post: Returns the largest value of the list
    assert(l!=NULL);

    node *p = l->next; //Make p point to the first empty element
    int max = 0; //Initialize max variable as 0

    while (p!=NULL){
        if (p->data > max) { //If the current data value is higher than one previously seen
            max = p->data;    //Update the max to the current data value
        }
        p = p->next; //update pointer
    }
    return max;
}
```