

Assignment Week 6

1)

Text answer

Old exam question

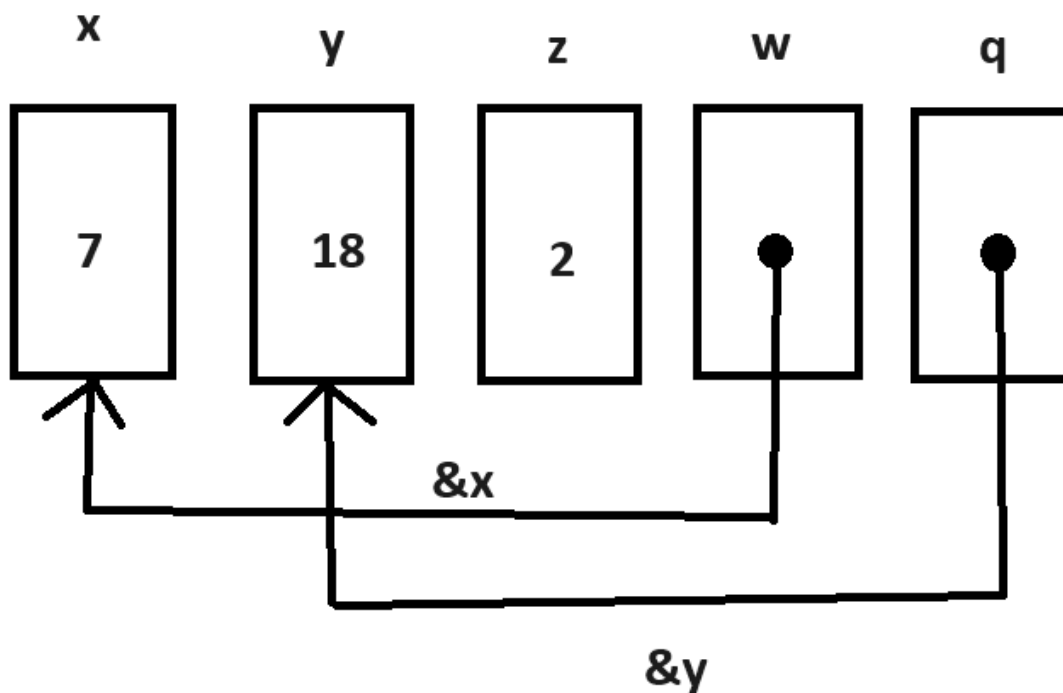
Consider the following program fragment:

```
1  int x;  
2  int y;  
3  int z;  
4  int* w;  
5  int* q;  
6  
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?

The program will print out "x=7, y=18, z=2". This can be seen in our diagram below:



3)

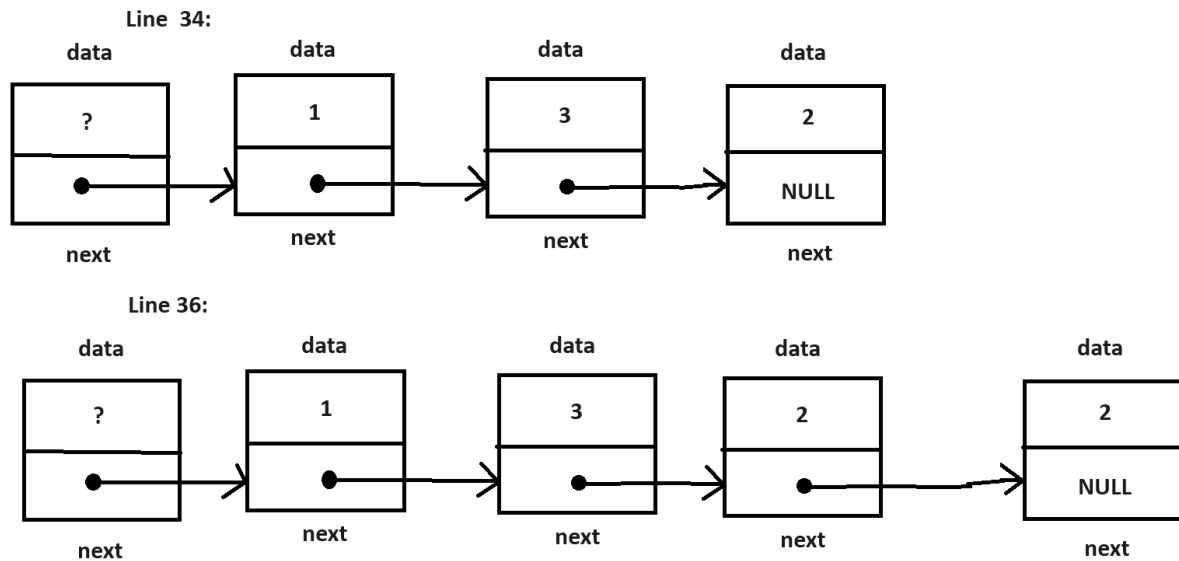
Text answer

Code answer

Consider the following program (you can find it in `./list.c`):

```
1  #include <stdio.h>
2  #include <assert.h>
3  #include <stdlib.h>
4
5  // NOTE: in the github repository this struct is defined in `./list.h`
6  typedef struct node {
7      int data;
8      struct node *next;
9  } node;
10
11 void add(node *head, int x) {
12     // pre: head points to the first, empty element.
13     //      The last element's next is NULL
14     // post: A new node containing x is added to the end of the list
15
16     assert(head != NULL);
17     node *p = head;
18     while (p->next != NULL) {
19         p = p->next;
20     } // p points to the last element
21
22     node *element = malloc(sizeof(node));
23     element->next = NULL;
24     element->data = x;
25     p->next = element;
26 }
27
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
37
38     return 0;
39 }
```

- (a) Draw **two** diagrams that shows the contents of `list` at the two lines with the comment `// show list here` in `main` (line 34 and 36).



(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 }

```

The code doesn't fulfill the post conditions. The postconditions state that the program must print out the values of the list, but this program only prints out the value of the first element in the list.

The program gets called with the parameter a pointer "l", which points to the node "head". It then declares a pointer "p", which points to the address that the variable "next", in the node that "l" points to, points to.

The while loop is run as long as the pointer "p" points towards something that isn't NULL. Here the program doesn't fulfill the precondition. Because if we look, at precondition, it tells us, that the last elements "next" is NULL, not the last element is NULL, så if the program should be correct, we should change the condition in the while loop:

while (p != NULL){	→	while (p->next != NULL){
...		...
}		}

There is also a problem in the while loop, because we get the variable "data" from the node that "p" points towards. But we don't change what "p" points towards, so if the condition for the while is true when we enter the while loop the first time, then it

will be true forever, and the while loop will never stop. To make the program correct, we would add the line:

```
p = p->next;
```

So now “p” points towards what the “next”, that p pointed to before , pointed to.