

## Aflevering 6 - Programming with pointers

1)

Outputtet bliver:

$x = 7$   $y = 18$   $y = 2$

2)

```
int max(int* numbers, int size)
{
    assert(size>0);

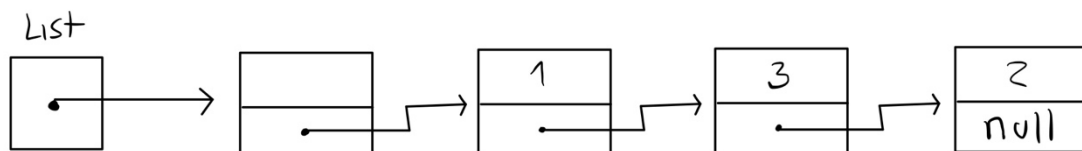
    int maxvalue = numbers[0]; // gemmer størsteværdien i array på første plads i arrayet.

    for (int i = 1; i < size; i++) { // loop der tjekker største værdien i et array
        if (numbers[i] > maxvalue)
            maxvalue = numbers[i];
    }
    return maxvalue;
}
```

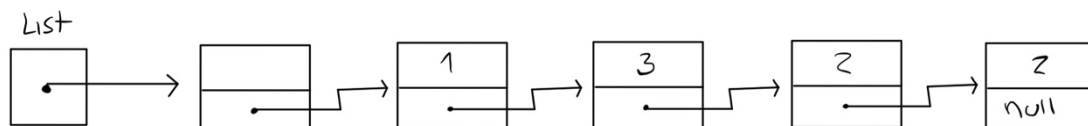
3)

a)

Listen efter vi har call'et funktionen 3 gange.



Efter sidste gang vi call'er funktionen



b)

```
int size(node *l){
    node *p = l; // Pointer som peger på node elementets *next
    int i = 0;

    while (p->next != NULL) { // Kører loop ind til next = NULL
        p = p->next; // sætter p til at pege på node elementets next
        i++; //tæller længden af listen, selvom første tomme element bliver talt med går det op, da sidste element ikke bliver talt med
    }
    return i;
}
```

c)

Først bliver der defineret en node som er en pointer ved navn p, pointeren peger på L nodens next. Derefter kører while loopet for evigt, da p aldrig bliver opdateret og dermed forbliver ikke lig med NULL. Derfor bliver p nodens data som er en integer printet uendeligt.

d)

```
void printout(node *l) {  
    node *p = l->next; // laver en pointer p, som peger på første ikke tomme element  
  
    while (p!=NULL){  
        printf("%d, ",p->data); // Printer data værdierne  
        p = p->next; // Pointer peger nu på elementets next  
    }  
    printf("\n");  
}
```

e)

```
int largest(node *l){  
    /*  
    pre: head peger på det første tomme element. sidste elements next er NULL, size(l>0)  
    post: returnerer den største værdi i listen */  
  
    node *p = l->next; // Laver en pointer på det første tomme element  
    int result = p->data; // variabel til at gemme resultaterne  
  
    while(p->next!=NULL) { // Kører loopet indtil sidste element  
        if (p->data > result) {  
            result = p->data; // Opdaterer result til nuværende største værdi  
        }  
        p = p->next; // Opdaterer p til næste element  
    }  
  
    return result;  
}
```