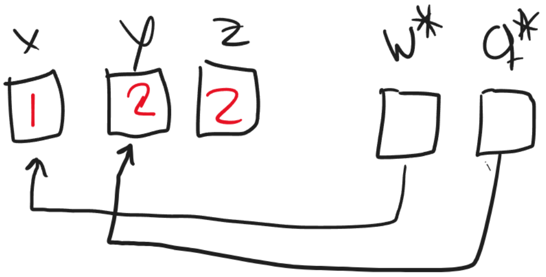
1)

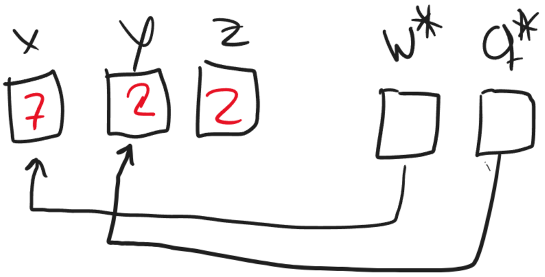
Text answer Old exam question Consider the following program fragment:

```c
int x;
int y;
int z;
int* w;
int* q;

x = 0;
y = 1;
z = 2;
w = &x;
q = &y;
*w = y;
*q = z;
*w = x + y + z + *q;
*q = x + y + z + *w;

printf("x=%d, y=%d, z=%d\n", x, y, z);
```

? Question

What does the program print when it is executed?



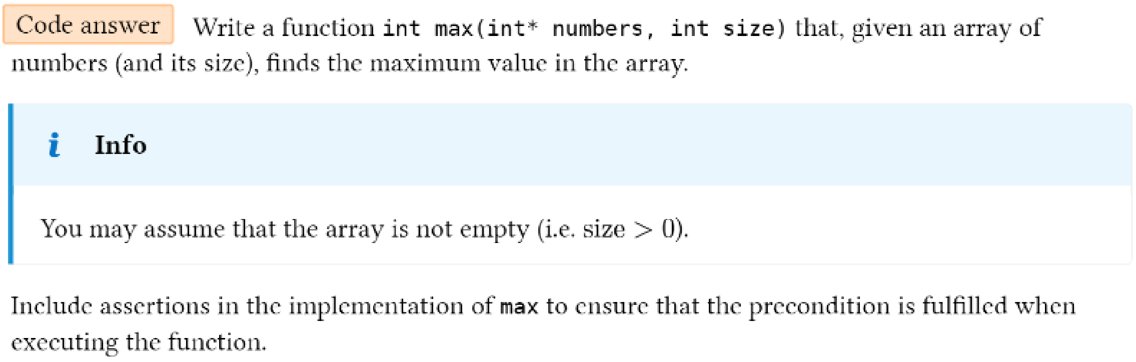


x = 1+2+2+2 = 7

y = 7+2+2+7=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.

> *i* **Info**
>
> 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.

https://prod.liveshare.vsengsaas.visualstudio.com/join?7C18791E55FD9B603C8A00D34639A068E493

3)

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

```c
#include <stdio.h>
#include <assert.h>
#include <stdlib.h>

// NOTE: in the github repository this struct is defined in `./list.h`
typedef struct node {
  int data;
  struct node *next;
} node;

void add(node *head, int x) {
  // pre:  head points to the first, empty element.
  //       The last element's next is NULL
  // post: A new node containing x is added to the end of the list

  assert(head != NULL);
  node *p = head;
  while (p->next != NULL) {
    p = p->next;
  } // p points to the last element

  node *element = malloc(sizeof(node));
  element->next = NULL;
  element->data = x;
  p->next = element;
}

int main() {
  node *list = malloc(sizeof(node));
  list->next = NULL; // create first empty element
  add(list, 1);
  add(list, 3);
  add(list, 2);
  // Show list here
  add(list, 2);
  // Show list here

  return 0;
}
```

(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).

> ***i*** **Info**
>
> The first element is empty and holds no data. I.e. if I have a list with two elements, it has $2 + 1 = 3$ 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. does the code fulfill the post condition? If not, what happens?)

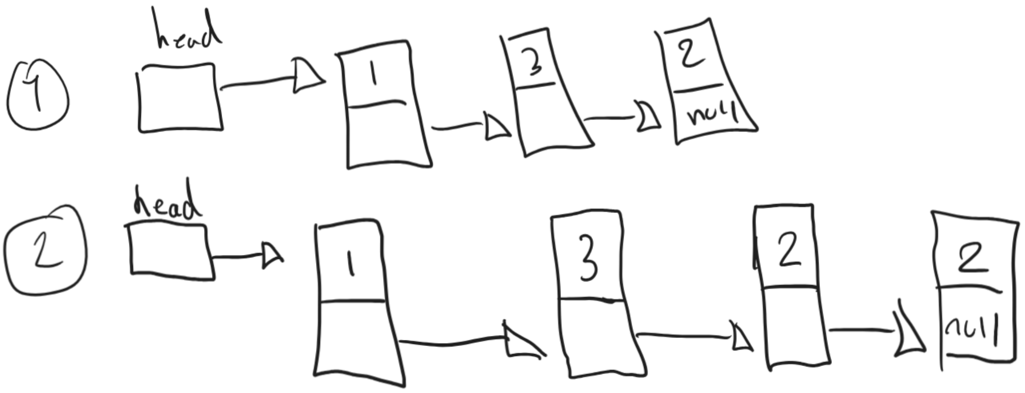
```c
void printout(node *l) {
  // pre:  head points to the first, empty element.
  //       The last element's next is NULL
  // post: The values of the list are printed out
  node *p = l->next;
  while (p != NULL) {
    printf("%d, ", p->data);
  }
  printf("\n");
}
```

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

(e) Write a function `int largest(node *l)`. The pre- and post conditions are the following:

```c
// pre:  head poinst to the first, empty element.
//       The last element's next is NULL.
// post: Returns the largest value of the list
```

a)


1
head
1
3
2
null
2
head
1
3
2
2
null