

Aufgabe 1):

Teil b)

Es gilt 133 Bytes Speicherlöcher zu finden.

In `main.c:41` belegt `foo` 23 Bytes, welche nie freigegeben werden.

Die Restlichen 110 Bytes entstehen in `main.c:54`,

```
allocated = leaking_function(bar, bar+5);
```

`leaking_function(2)` funktioniert folgendermaßen, wenn die grenzen eines validen char-arrays eingegeben werden (andere Fälle treten hier nicht auf und werden dementsprechend nicht beachtet):

Für jeden numerischen `n` Wert im Array:

Für `n = 0`, tue nichts

Für `n > 0`, alloziere `n*sizeof(char)` bytes Speicherplatz.

Nun gebe man den pointer für ausschließlich den letzten allozierten Speicher zurück.

Klar ist: Für die allozierten Blöcke wird, ausgenommen den letzten, keine Referenz gespeichert, wodurch sie nicht freigegeben werden können.

In `bar` sind dies die Werte 11, 22, 33, 44 - aufsummiert 110.

Somit sind alle 133 Bytes erklärt.