Programmierung:

Haben Hauptspeicher als char-Array
zuerst print-fkt zum

Variante 1 (elegant):

Haben RAM-char-Array. Das wird aufgeteilt in Blöcke.
Erste Blocks sind Metadaten (steht größe und Pointer drin).

Alternierend Metadaten-Block und eigentlicher Speicherblock.
mem_allocate wird neuen metadatenblock

ganz hinten auch noch Block. Der zeigt auf den ersten.
Metadatenblock ist tricky. Der hat größe und zeigt auf den nächsten.

```
#define MEM_SIZE 256
char memory[MEM_SIZE];

struct mem_block{
        size_t size;
        struct mem_block *next;
};
```

Struct in Char-Arr bekommen?

```
struct mem_block *first = (struct mem_block *) memory;
// tun so als ob Startadresse vom char-array struct wäre.

first->size = 0;
first->next = NULL;

alloc(5)

first->size = 5;

// Wollen nun Adresse nach dem Metadatenblock
void *addr = ( (char *) first) + sizeof(struct mem_block);
ODER
void *addr = first + 1;          // Pointerarithmetik
```

// NUN auf das vom speicher auf nächsten Metadatenblock
wieder zu char_Array casten udn +1 casten

Variante 2 (billig)

Teilen Array in 10 feste groesse ein
ganz oben Metablock wo man dann sagt welche Position belegt ist 1001010101001

```
#define BLOCK_COUNT 256
```

```
#define BLOCK_SIZE 10

        // Verwaltung      Speiche
char memory[BLOCK_COUNT + BLOCK_COUNT * BLOCK_SIZE]
memory[0] = 1;          // Block 0 ist vergeben.
memory[9] = 0;          // Block 9 ist frei
```

und dann schauen ob allocate(n) größer ist als block groß. dann fehler.
--> super hohe interne Fragmentierung

- irgendwie merken, dass der block vergeben ist.

malloc aber nur für Array benutzen!!!