P5 specific allocator design

Heap block

block header

```
31 1 0

block payload size + padding size 0 a
pointer to next block.
```

payload

padding (optional)

if a = 1 \Rightarrow allocated
if a = 0 \Rightarrow free

\[
\text{block size} = \text{payload size} + \text{padding size}
\]

\[
\text{block header size} = 8 \text{ bytes (4 bytes for blocksize, 4 bytes for next ptr)}
\]

\[
\text{min block size} = 8 + 4 = 12 \text{ bytes}
\]
1. **Free block Organization**

```c
typedef struct block_hd {
    struct block_hd *next;
    int size_status;
} block_header;

block_header *list_head = NULL;
```

2. **Placing allocated blocks**

Placement policy = best fit.

`malloc(s)` → Starting from the first block: choose the smallest free block that fits `s` bytes.
Example for best fit

malloc (4)
The block that should be chosen is b4!
:: b4 fits 4 bytes more closely than b2.

3. Splitting free blocks

malloc (4) will split the above free block as shown:

malloc (8) will NOT split the block.
4. Coalescing free blocks

Case 1: next block is free, prev block is allocated.

```
f free(p);
```

Case 2: prev. block is free, next block is allocated.