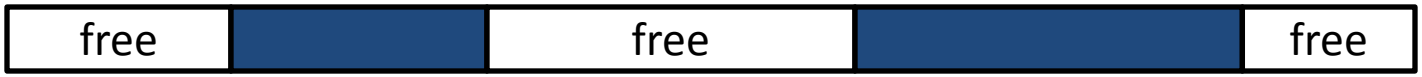
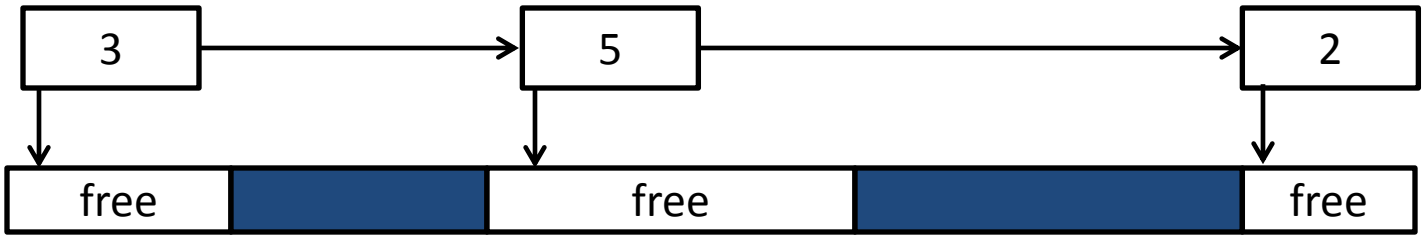
You are given a block of memory with several free and occupied blocks in it:



Your first task is to create a linked list of nodes with a) a pointer to the beginning of each free memory block and b) the length of the block:



The second task is to defragment the memory into a single block of free memory and a single block of occupied memory, preserving all data, using the linked list for iteration:

Note: free memory is represented by 0.
Note: pieces of memory are not necessarily bytes (in this example they are integers), but are bitwise copyable.
Note: you cannot use any STL containers.

```
// Insert code here
```

```
void main()
{
    int array[] = {
         0,   0,   0,              // free block of 3
        'C', 'O', 'N', 'T', 'I',   // occupied block of 5
         0,   0,   0,   0,   0,    // free block of 5
        'G', 'U', 'O', 'U',        // occupied block of 4
         0,   0,                   // free block of 2
        'S', '!' };                // occupied block of 2

    MemoryManager<int> mm(array, 21);
    mm.print();
    mm.defragment().print();
}
```

Expected output:
Free block lengths: 3, 5, 2 | Occupied block contents: CONTI, GUOU, S!
Free block length: 10 | Occupied block contents: CONTIGUOUS!