

Problem 1

False fragmentation is a phenomenon where there is a lot of available free memory chopped up into small, unusable free blocks.

The common ways are immediate coalescing and deferred coalescing. Immediate coalescing merges any adjacent blocks each time a block is freed. While deferred coalescing waits to coalesce free blocks at some later time.

Problem 2

Example code:

```
static void *find_fit(size_t asize)
{
    /* Best fit search */
    void *bp;
    void *best = NULL;
    unsigned int min = -1 ;
    for (bp = heap_listp; GET_SIZE(HDRP(bp)) > 0; bp = NEXT_BLK(P(bp))) {
        if (!GET_ALLOC(HDRP(bp)) && (asize <= GET_SIZE(HDRP(bp)))) {
            if (GET_SIZE(HDRP(bp)) == asize)
                return bp;
            if (GET_SIZE(HDRP(bp)) - asize < min ){
                min = GET_SIZE(HDRP(bp)) - asize;
                best = bp;
            }
        }
    }
    return best;
}
```

Problem 3

1)

16/1(p3)	16/1(p3)	40/0	40/0	8/1(P2)	8/1(P2)	24/0	24/0
----------	----------	------	------	---------	---------	------	------

16/1(P3)	16/1(P3)	24/1(P4)	24/1(P4)	16/0	16/0	8/1(P2)	8/1(P2)	24/1(P5)	24/1(P5)
----------	----------	----------	----------	------	------	---------	---------	----------	----------

2)

$$(16-4) + (24-10) + (8-0) + (24-9) = 49$$

$$\text{Or } (16-4) + (24-10) + (24-9) = 41$$

