Homework 12

Understanding Memory Performance

1. Consider the following function to copy the contents of one array to another:

```
void copy_array(long *src, long *dest, long n)
{
    long i;
    for (i = 0; i < n; i++)
        dest[i] = src[i];
}
```

Suppose `a` is an array of length 1000 initialized so that each element `a[i]` equals `i`.

(a) What would the array become if call `copy_array(a+1, a, 999)`?

(b) What would the array become if call `copy_array(a, a+1, 999)`?

(c) Our performance measurements indicate that the call of part a has a CPE of 1.2, while the call of part b has a CPE of 5.0. To what factor do you attribute this performance difference?

(d) What performance (CPE) would you expect for the call `copy_array(a, a, 999)`? Please explain your answer.

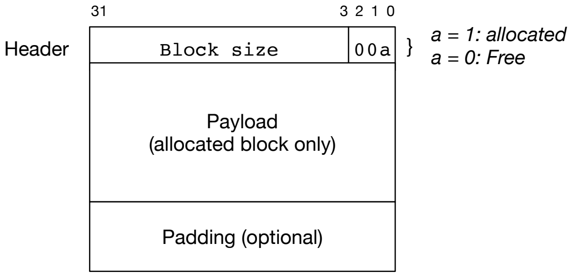
Dynamic Memory Allocation

1. Determine the block sizes and header values that would result from the following sequence of `malloc` requests.

Request	Block size (decimal bytes)	Block header (hex)
`malloc(1)`	____	____
`malloc(5)`	____	____
`malloc(12)`	____	____
`malloc(13)`	____	____

Assumptions:

- *words* are 4-byte objects and *double words* are 8-byte objects.
- The memory allocated to the user is at the granularity of *word*. That is, the size requested are rounded up to the nearest multiple of 4 bytes.
- The allocator maintains double-word alignment and uses an imiplicit free list with the block format as below.



2. Determine the minimum block size for each of the following combinations of alignment requirements and block formats. Assumptions: implicit free list, zero-size payloads are not allowed, and headers and footers are stored in 4-byte words.

Alignment	Allocated block	Free block	Minimum block size (bytes)
Single word	Header and footer	Header and footer	____
Single word	Header, no footer	Header and footer	____
Double word	Header and footer	Header and footer	____
Double word	Header, no footer	Header and footer	____