on pointer arithmetic

```c
char  bufch[12];
char  *pch;
pch = bufch;
pch++;  
```

If `bufch` is at address 8000, then `pch` will be 8001.

( `pch + 1 * size of char` )
int bufint[12];
int *pint;
pint = bufint;
pint++;

If bufint is at address 8000, then pint will be 8004.

(pint + 1 * size of int)
struct node {
    int value;
    struct node *next;
} Node;
Node *ptemp;

ptemp = malloc( sizeof (Node) );
ptemp++;
applying this to your programming assignment:
we want to use our heap space as both \texttt{block\_header} structures and as a bunch of contiguous bytes.
Given `ptr`, you want to return (calculate) the address of `block_header`.
Assume declaration is

```c
block_header *ptr;
```

`ptr + 1` is `(ptr + 1 * sizeof block_header)`
OR, with

```
block_header *ptr;
```

the expression is `( (char *) ptr ) + 8`
Contrast this with *(incorrect)*:

```c
block_header *ptr;
```

the expression  `ptr + 8`