CSE 484 In-section Worksheet #3

Q1. Which `gdb` command allows us to:

view the four words starting at `ebp` in hex? x/4xw $ebp

view the next five instructions at `eip`? x/5i $eip

view all instructions for function `foo`? disas foo

Q2. Which register does the x86 instruction `RET` affect? How, exactly?

"The ret instruction implements a subroutine return mechanism. This instruction first pops a code location off the hardware supported in-memory stack (see the pop instruction for details). It then performs an unconditional jump to the retrieved code location."
https://www.cs.virginia.edu/~evans/cs216/guides/x86.html

Q3. What do `tmalloc()` and `tfree()` do?

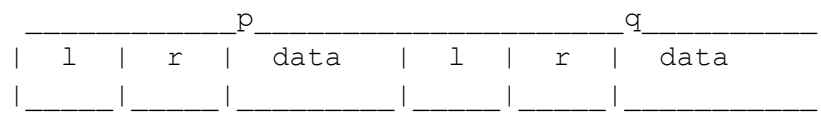
See slides page 6

Q4. What's the issue with this code?

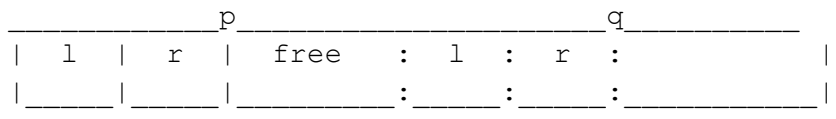
```
      char *p; char *q;
      if ( (p = tmalloc(128)) == NULL)
      { exit(EXIT_FAILURE); }
      if ( (q = tmalloc(128)) == NULL)
      {exit(EXIT_FAILURE); }
A
      tfree(p);
      tfree(q);
B
      if ( (p = tmalloc(256)) == NULL)
      {exit(EXIT_FAILURE); }
      obsd_strlcpy(p, arg, 256);
C
      tfree(q); Double free!
```

Q5. Based on `tmalloc.c`, draw what the heap/free list looks like at points, A, B, and C. Include chunk structure and label `p` (at or before point B), `p` (at point C), and `q`. Where is `buf` copied?

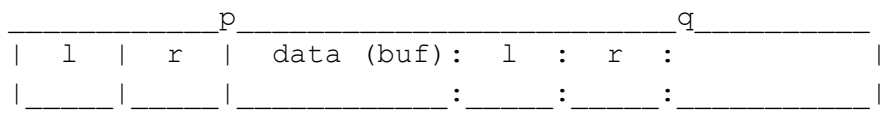
A:



B: (think of the left and right divisions of q as implicit because of consolidation)



C:



Q6. Given your diagrams and the following code for chunk consolidation (from `tmalloc.c`), what do the following statements do when executed in the call `tfree(q)` after point C?

```
q->s.r = p->s.r;
p->s.r->s.l = q;
```

slides page 10:
if we control chunks p (and q), this code will write the value of q (address of buffer?) to a location we specify (location of saved EIP?).