Quiz 0
Solutions

Answers other than the below may be possible.

Multiple Choice.

0. c
1. b
2. b
3. c
4. d

True or False.

5. F
6. F
7. T
8. T

Short Answers.

9.
```
Repeat n times:
    Let count = 0.
    For each element i:
        If element i and its neighbor are out of order:
            Swap them.
            Increment count.
    If count == 0:
        Break;
```

10. Using only global variables increases the risk that some functions might change their values unbeknownst to other functions. It's particularly risky to reuse (popular) symbols (*e.g.*, `i`). Using only global variables also tends to make code less readable if variables' declarations are no longer proximal to lines of code that utilize those variables.

11. On each iteration (of the loop induced by lines 3 and 4), half of the students sit down. A set of size *n* can only be halved log *n* (or, more precisely, $\log_2 n$) times.

12. Because both 5 and 9 are integers, `5/9` evaluates to `0`, so the entire expression becomes `0.0`.

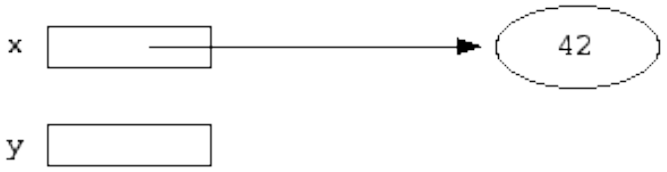
```
float c = (5/9.0) * (f - 32);
```

13. `sscanf` returns the number of successfully matched and assigned input values (*i.e.*, format codes). If a user provides an `int` (surrounded, perhaps, by whitespace) followed by one or more non-numeric `char`s, the first of those `char`s will end up in `c`, the result of which will be a return value of `2` (*i.e.*, not `1` as desired). If a user provides one or more non-numeric `char`s (surrounded, perhaps, by whitespace) before any numeric input, `sscanf` will return `0` (*i.e.*, not `1` as desired). Only if a user provides an `int` and only an `int` (surrounded, perhaps, by whitespace) will `sscanf` return `1` as desired.

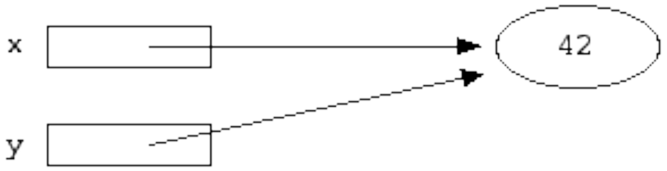
Pointer Fun with Binky.

14.

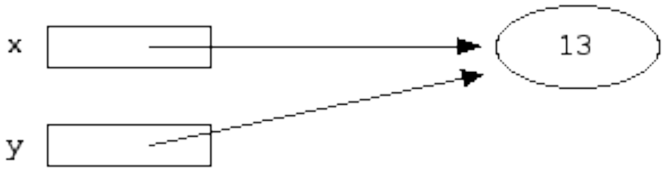
```
*x = 42;
```



```
y = x;
```

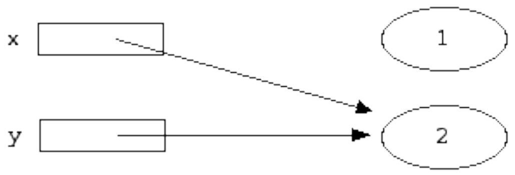


```
*y = 13;
```



15.

```
int *x = malloc(sizeof(int));
int *y = malloc(sizeof(int));
*x = 1;
*y = 2;
x = y;
```



sizeof.

16.

`sizeof(char)`	1
`sizeof(char *)`	4
`sizeof(int)`	4
`sizeof(int *)`	4
`sizeof(long)`	4
`sizeof(long long)`	8
`sizeof(float)`	4

Rapid Fire.

17. `GetString` allocates memory dynamically via `malloc`. If the caller of `GetString` does not subsequently release that memory via `free`, memory has been leaked.

18. `switch` can facilitate readability, since it's sometimes easier to skim a list of cases without the additional overhead of syntax required by `if`, `else if`, and `else` (*e.g.*, parentheses, repetition of variables names, and `==` itself). But, choice of `switch` over `if`, `else if`, and `else` is sometimes simply a matter of style (*i.e.*, personal preference).

19. Vigenère's cipher allows for *n*-letter keys, the result of which, for a 26-letter alphabet, is a total of 26^n possible keys. Caesar's cipher, meanwhile, assuming the same alphabet, allows for only 26 keys. Vigenère's cipher is said to be strong thatn Caesar's because an adversary would have to try many more keys for it than for Caesar's to crack (*i.e.*, guess) someone's key.

20. Buffer overflow attacks can be prevented by ensuring that arrays' boundaries cannot be exceeded by writes (as by checking the length of every input against the length of its destination in memory).

Swap Time.

21.
```
void
swap(student *a, student *b)
{
    if (a != NULL && b != NULL)
    {
        char *tmp = a->house;
        a->house = b->house;
        b->house = tmp;
    }
}
```

"GCC hates me."

22. Doug has forgotten

```
#include <stdio.h>
```

atop his code. Inserting that line should resolve.

23. Michelle has forgotten to link her code against CS 50's library. Recompiling with `-lcs50` should resolve.

24. Rose might have dereferenced a `NULL` or otherwise invalid pointer, might have exceeded the bounds of some array, might have made too many recursive function calls, or, more generally, might have touched memory that did not (yet) belong to her program.

Two Lists.

25.
```
typedef struct node
{
    struct node *prev;
    int n;
    struct node *next;
}
node;
```

26.

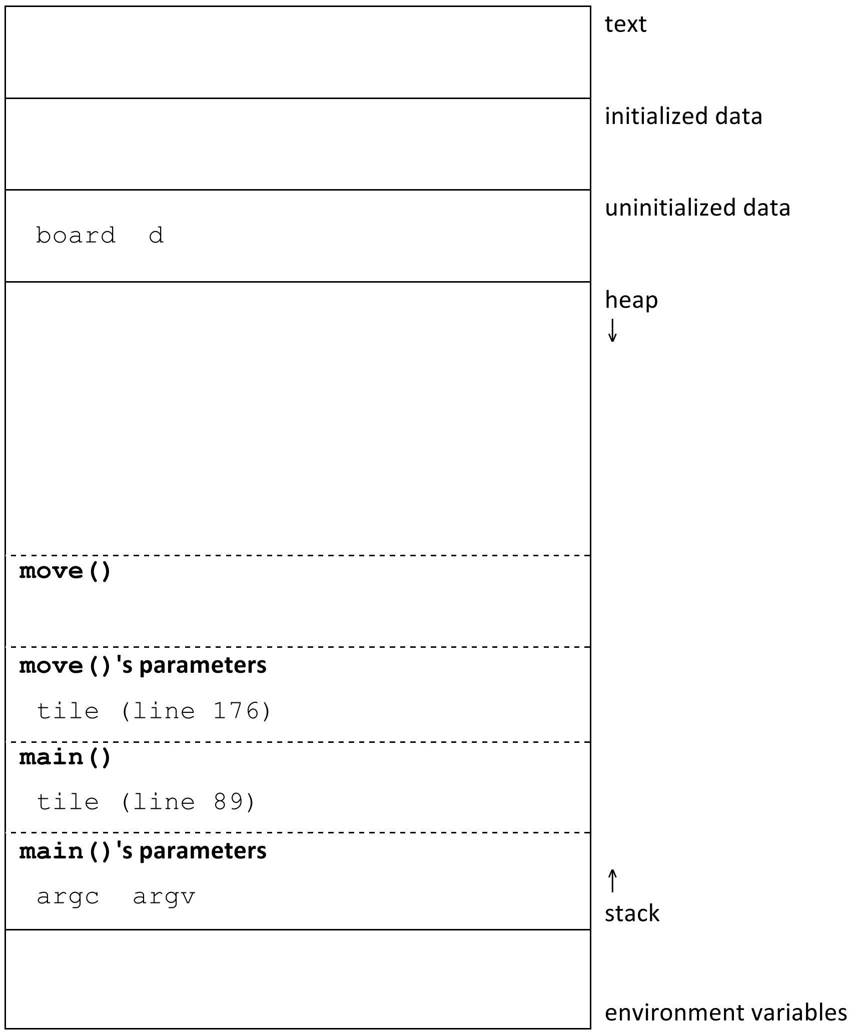
Algorithm	***O***	**Ω**
inserting into a **sorted** singly linked list	$O(n)$	$\Omega(1)$
searching a **sorted** singly linked list	$O(n)$	$\Omega(1)$
sorting a **sorted** array with Selection Sort	$O(n^2)$	$\Omega(n^2)$
sorting an **unsorted** array with Merge Sort	$O(n \log n)$	$\Omega(n \log n)$

C meets PHP.

27.
```
char *
ucwords(const char *s)
{
    if (s == NULL)
        return NULL;

    int n = strlen(s);
    char *t = malloc((n+1) * sizeof(char));
    if (t == NULL)
        return NULL;

    bool first = true;
    for (int i = 0; i <= n; i++)
    {
        if (first)
        {
            t[i] = toupper(s[i]);
            first = false;
        }
        else
        {
            t[i] = s[i];
            if (s[i] == ' ')
                first = true;
        }
    }

    return t;
}
```

Distant Memory.

28.



Extra Noncredit.

29. Try it![*]

[*] `http://en.wikipedia.org/wiki/Quine_(computing)`