

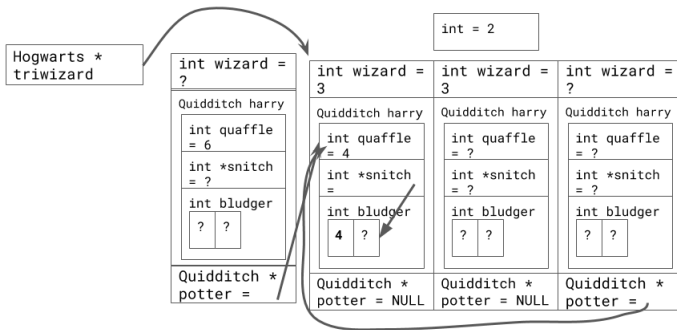
Pointers and Linked Lists: Solutions

1. Pointer Mystery

line 1: 70 25 line 2: 70 70

2. Pointer Trace

Stack (gryffindor): Heap:



3. Stretch (Arrays)

```
void ArrayIntList::stretch(int k) {
    if (k <= 0) {
        // remove all elements from the list
        mysize = 0;
    } else {
        // resize array as needed to fit
        if (capacity < mysize * k) {
            int* bigger = new int[k * mysize]();
            for (int i = 0; i < mysize; i++) {
                bigger[i] = elements[i];
            }
            delete[] elements;
            elements = bigger;
            capacity = k * mysize;
        }

        // stretch the elements
        for (int i = mysize - 1; i >= 0; i--) {
            for (int j = 0; j < k; j++) {
                elements[i * k + j] = elements[i];
            }
        }
        mysize *= k;
    }
}
```

4. Mirror (Arrays)

```
void ArrayIntList::mirror() {
    if (mysize * 2 > capacity) { // grow array to fit
        int* bigger = new int[2 * capacity];
        for (int i = 0; i < mysize; i++) {
            bigger[i] = elements[i];
        }
        delete[] elements;
        elements = bigger;
        capacity *= 2;
    }
    for (int i = 0; i < mysize; i++) {
        elements[2 * mysize - 1 - i] = elements[i];
    }
    mysize *= 2;
}
```

5. Date (Classes)

Date.h:

```
#ifndef _date_h
#define _date_h

class Date {
public:
    Date(int m, int d);
    int daysInMonth() const;
    int getDay() const;
    int getMonth() const;
    int nextDay();
    int toString() const;

private:
    int month;
    int day;
};

#endif
```

Date.cpp:

```
#include "Date.h"

Date::Date(int m, int d) {
    month = m;
    day = d;
}

int Date::daysInMonth() const {
    if (month == 4 ||
        month == 6 ||
        month == 9 ||
        month == 11) {
        return 30;
    } else if (month == 2) {
        return 28;
    } else {
        return 31;
    }
}

int Date::getDay() const {
    return day;
}

int Date::getMonth() const {
    return month;
}
```

```
void Date::nextDay() {
    day++;
    if (day >= daysInMonth()) {
        month++;
        day = 1;
        if (month > 12) {
            month = 1;
        }
    }
}

string Date::toString() const {
    string result = "";
    if (month < 10) {
        result += "0";
    }
    result += integerToString(month);
    result += "/";
    if (day < 10) {
        result += "0";
    }
    result += integerToString(day);
    return result;
}
```