Announcements

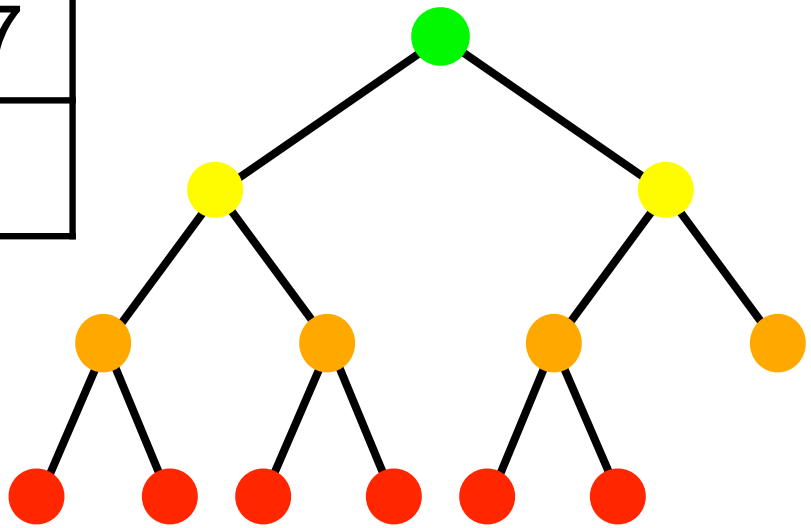
MP3 available, due 9/30, 11:59p. EC due 9/23.

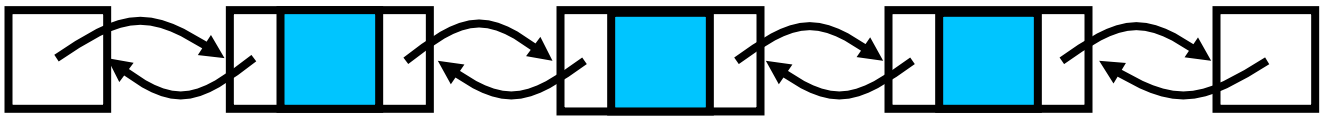
Exam2: 9/25-9/28

Last time, function templates.

Classes can be given templates too:

0	1	2	3	4	5	6	7





Class templates:

```
template <class T>
class ezpair {
private:
   T a, b;
public:
   ezpair (T first, T second);
   T getmax ();
};
```

```
template <class T>
T ezpair<T>::getmax() {
   T retmax;
   retmax = (a>b ? a : b);
   return retmax;
}

template <class T>
ezpair<T>::ezpair(T first,T second){
   a = first;
   b = second;
}
```

```
int main () {
   ezpair<int> twoNums(100, 75);
   cout << twoNums.getmax();
   return 0;
}
```

Class templates:

```
template <class T>
class ezpair {
private:
   T a, b;
public:
   ezpair (T first, T second);
   T getmax ();
};
```

```
template <class T>
T ezpair<T>::getmax() {
   T retmax;
   retmax = (a>b ? a : b);
   return retmax;
}

template <class T>
ezpair<T>::ezpair(T first,T second){
   a = first;
   b = second;
}
```

Challenge1: write the function signature for the copy constructor (if we needed one) for this class.

_____ _____ :: _____ (_____)

Challenge2: How do you declare a dynamic array of `mypair`s of integers?

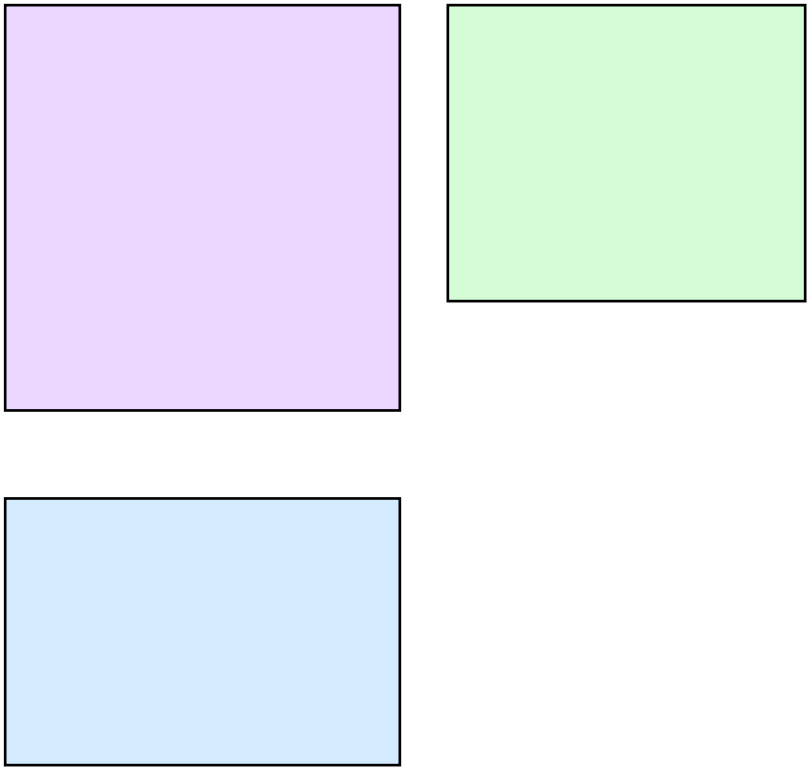
Challenge3: How do you allocate memory if you want that array to have 8 elements?

A note on templates:

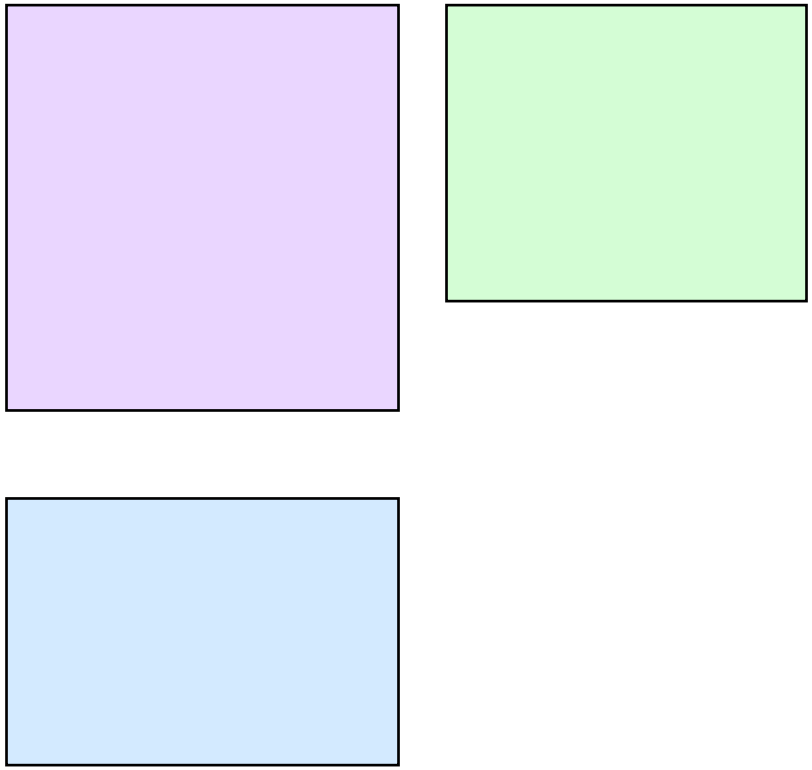
```
template <class T, class U>
T addEm(T a, U b) {
   return a + b;
}

int main() {
   addEm<int,int>(3,4);
   addEm<double,int>(3.2,4);
   addEm<int,double>(4,3.2);
   addEm<string,int>("hi",4);
   addEm<int,string>(4,"hi");
}
```

Template compilation:

Old:

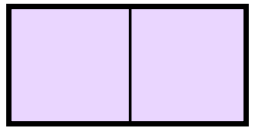


New:



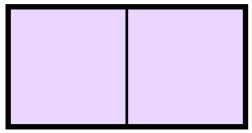
Toward a new memory model:

```
struct listNode {
   LIT data;
   listNode * next;
   listNode(LIT newData):data(newData),next(NULL){}
};
```

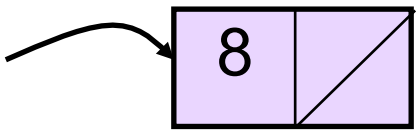


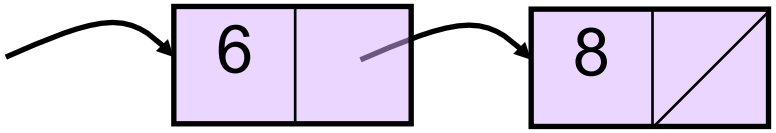
What is the result of this declaration?

```
listNode<int> nln(5);
```

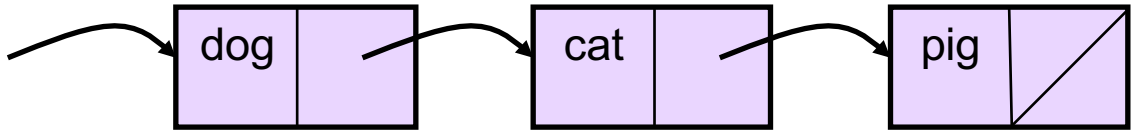


Write code that would result in each of these memory configurations?





Example 1: `insertAtFront<farmAnimal>(head,cow);`



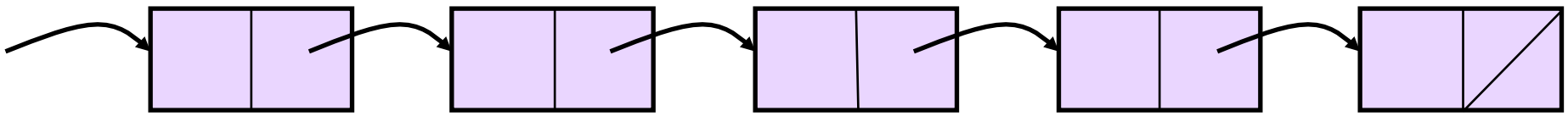

```
void insertAtFront(listNode *  curr, LIT e) {

}
```

Running time?

```
struct listNode {
   LIT data;
   listNode * next;
   listNode(LIT newData):data(newData),next(NULL){}
}
```

8 4 2 6 3

Example 2:



```
void printReverse(listNode *  curr) {
```

```
}
```

Running time?

```
struct listNode {
   LIT data;
   listNode * next;
   listNode(LIT newData):data(newData),next(NULL){}
}
```