Write a code segment that will create a double called number, a pointer to a double called ptr, and an array of 10 doubles. Make ptr point to number and then assign the largest value from the array to number without using the "number" label (i.e., through the pointer).

Recall the Rational class that we have been developing in lecture:

```
#ifndef RATIONAL_H
#define RATIONAL_H
#include <fstream>

class Rational
{
public:
  // Assume functions for class are still here



private:
  // Puts object in reduced form
  void reduce();
  long int num;
  long unsigned int den;
};

std::ostream& operator<<(std::ostream& os, const Rational& rtnl);

#endif
```

Write whatever functions (add prototypes to class and implement below) you need to allow the following code to work:

```
Rational a(5,7);
Rational b(2,9);
std::cout << a - b << '\n';
```

Write the header file associated with a Time class that will allow Time objects to be declared and used as shown in the code below. Note: You should indicate all the appropriate preprocessor directives (#include files, etc). Write **only** the function **prototypes** for the functions **used in the code below**. You **do not** need to **implement** the functions.

```
#include <iostream>
#include "Time.h"

int main()
{
  Time appointment;
  const Time bedtime(22); // Sets time to 10pm (22:00:00 military time)
  std::cout << "Please enter the time of your appointment ";
  std::cin >> appointment;
  std::cout << "If your appointment lasts thirty minutes, it will be done at "
            << appointment + 30 << std::endl;
  int freeTime = bedtime - 30 - appointment;
  std::cout << "That will give you " << freeTime << " minutes until bedtime."
  return 0;
}
```

The following class has no data members. The class is used to group similar functions together. Both randomShuffle and pairwiseShuffle reorder the elements in the vector that is past to them. Each function just uses a different algorithm for doing the reordering.

```
#ifndef RANDOMIZEDORDER_H
#define RANDOMIZEDORDER_H

#include <vector>

class RandomizeOrder {
public:
  static void randomShuffle(std::vector<double>& data) const;
  static void pairwiseShuffle(std::vector<double>& data) const;
private:
  // No data members
};

#endif
```

Suppose a vector with 3,000 doubles in it called numbers exists. Show the function call that would reorder the numbers in numbers using the pairwiseShuffle function from the RandomizeOrder class.

Recall that the String class developed in lecture had two data members:

```
char* letters;
unsigned int sz;
```

Where letters == NULL when sz == 0. For any other value of sz, letters points to a character array that is sz big.
Implement the operator from the String class that is used on the line indicated below:

```
String name = "Jon ";
String last = "Dough";
name += last;  // <-- Implement the operator used on this line
```

Recall that the List<T> class developed in lecture had two data members:

```
unsigned int numEl;
Node<T>* start;
```

Where start == NULL when numEl == 0. For any other value of numEl, start points to the first Node<T> element in the list.
Implement the copy constructor from the List<T> class.

Consider the useless class below:

```
class Useless {
public:
  Useless();
  Useless(const Useless& org);
  virtual ~Useless();
  Useless& operator=(const Useless& rhs);
  virtual bool irrelevant() const = 0;
private:
  bool x;
};
```

Now consider the irrelevant member function.

(a) Describe the implications of the `virtual` keyword in front.
(b) Describe the implications to the function of the = 0 at the end.
(c) Describe the implications to the class of the = 0 at the end.
(d) What is the terminology used to describe such a function.