Prove that the following algorithm terminates using the method discussed in lecture and in the text. If the algorithm does not terminate, indicate an input condition for which the algorithm will not terminate.

```
unsigned int Number(unsigned int n)
{
  if(n<3) {
    return n;
  }
  return Number(n-2) + Number(n-1);
}
```

Give the asymptotic time complexity for the following algorithm using big-oh notation. Justify your answer with a short explanation.

```
unsigned int Number(unsigned int n)
{
  for(int i=n-5; i<n+5; ++i) {
    double pi = 3.1415;
  }
  if(n<3) {
    return n;
  }
  return Number(n-2) + Number(n-1);
}
```

Explain how it is possible to implement two stacks using one array. Your stacks should not generate an overflow condition unless the sum of the number of elements held in the stacks exceeds the size of the array.

Briefly describe how a set stores its data. How does this storage technique result in a time complexity of $O(\log n)$ for insert, erase, and find member functions?

Briefly describe the challenges associated with the design decision in the STL list class to have the end member function point to one past the end of the list. Discuss at least one alternative for implementing the list::iterator class such that the end iterator points to one past the end of the list.

Briefly describe how a vector could be adapted to represent a complete binary tree.

Briefly explain the differences between the following two member functions from the STL multimap class: lower_bound() and upper_bound().