



The `BigInt` class developed in class had the following constructor:

```
BigInt(const std::string& val, int level=1);
```

Implement this function with the additional constraint that an `assert` should be used to make sure the string passed in does not have any zeroes at the beginning (e.g., "0000123" is not allowed).



Implement the following member function for the `BigInt` class:

```
bool operator<(const BigInt& val) const;
```

```
{
  // Create a pointer to an unsigned integer , ptr1 , here

}

{
  // Create a pointer to an unsigned integer , ptr2 , here

}

{
  // Allocate space for the value 34567 and have ptr2 point to it

}

ptr2 = ptr 1;
}
// Deallocate the space for the value 34567 created above

}
```

Recall that the string class we discussed in lecture had two data members:

```
unsigned int sz;  
char* contents;
```

Rewrite the default constructor so that it allocates a character array of size zero and has `contents` point to it. **Also**, rewrite the destructor so that it assumes that a character array is allocated, even if `sz==0`.



Recall that the `List<T>` class we discussed in lecture had two data members:

```
long unsigned int sz;  
T* start;
```

and that the `Node<T>` class we discussed in lecture had two data members:

```
T data;  
Node<T>* next;
```

Write the `push_front` member function for the `List<T>` class.

Indicate any syntax errors with the following code:

```
DesktopItem* ptr = new DesktopItem("file");
std::cout << ptr.getName();
std::cout << ptr->size();
delete ptr;
ptr = new TextDoc("file.txt");
std::cout << ptr->getName();
std::cout << ptr->size();
TextDoc txtDoc("file2.txt");
Folder fldr("folder");
DesktopItem dtItem("item");
txtDoc = fldr;
dtItem = fldr;
fldr = dtItem;
```

Explain the differences between virtual and non-virtual functions. Give one code example where being virtual matters and one code example where being virtual does not matter.

Quizzes



Name:

Explain the effect the designation `const`, `static`, and `friend` have on member functions.