

```
class longnumber {
- 123567890123567890123567890123567890123567890123567890123567890
//Must be able to represent + or - number (int) of infinite size
}
```

```
longnumber l1(1986)
longnumber l2("56789156789") ;
```

```
012345678901234567890123456789012345678901234567890123456789012
```

```
const char* n = "- 123567890123567890123567890123567890123567890123567890123567890"
```

```
is_negative() = true
is_positive() = false
num_digits() = 63
l[0] = 1 l[1] = 1 l[62] = 0
```

```
cout << l2(put eoln at after 40 digits
if (l1 == l2)
if (l1 != l2)
```

```
overload +
Guaranteed l1 and l2 is >=0
```

```
l1(1986)
l2(45)
longnumber a = l1 + l2
```

```
l1("45678")
l2(' '45689145")
longnumber a = l1 + l2
```

```
overload *
l1 is integer or longnumber
can be negative
l2 is GUARANTEED to be int >=0
```

```
l1(1986)
l2(45)
longnumber a = l1 * l2
```

```
l1("45678")
l2(23)
longnumber a = l1 * l2
```

**Factorial**

```
n is int >=0. n =10, 100, 1000, 10000
longnumber a = longnumber::fact(n);
```