

6. (30points) In our Project #2, class `inf_int` is defined as follows. It can represent an arbitrarily big integer number in practice and the limit it can represent is maximized.

In (1), (2), and (3), Do not call other member functions. Instead, You may use standard library in your code.

Your code should manage the memory correctly and efficiently. Your code should be grammatically and logically correct.

```
-----  
class inf_int  
{ private:  
    char* digits;           // points to a string of digits. Perform dynamic allocation when necessary.  
    unsigned int length;    // stores the number of actual digits  
    bool thesign;          // we assume that thesign is false if negative integer, true otherwise.  
    // ex) 153111111111111111 : digits -> "11111111111111351", length=17, thesign=true;  
    // ex) -12345555555555 : digits -> "55555555554321" , length=14, thesign=false  
  
public :  
    inf_int();              // constructor. assign 0 as a default value.  
    inf_int(int n);        // constructor. the input integer n is converted to inf_int format.  
    inf_int(const inf_int& x); // copy constructor  
  
    // other member functions should be here but they are just not shown.  
};  
-----
```

(1) Write C++ code for following constructor.

```
inf_int::inf_int() // assign 0 as a default value
```

```
{  
  
}  
}
```

(3) Write C++ code for following constructor.

```
inf_int::inf_int(int n)
```

```
{  
  
}  
}
```

(2) Write C++ code for following constructor.

```
inf_int::inf_int(const inf_int& x) // copy constructor
```

```
{  
  
}  
}
```