

```

1  #ifndef __BigInt_
2  #define __BigInt_
3
4
5  struct BigInt
6  {
7      char* digits; // no leading zeros, doesn't end with '\0'
8      int n_digits;
9  };
10
11 BigInt BigInt_create(const char* s); //create from C string.
12                                     //The string is NULL terminated and
                                     contains
13                                     //only characters between '0' to '9'.
14
15 void BigInt_print(const BigInt& bi); //print number to the screen
16
17 BigInt BigInt_create(int num); //creat from int
18
19 int BigInt_compare(const BigInt& a, const BigInt& b); //-1 if a<b
20                                     // 0 if a = b
21                                     // 1 if a>b
22
23
24 void BigInt_assign(BigInt& a, const BigInt& b); // create copy of b in a
                                     (like a = b; in integers)
25                                     // a <- b
26
27
28 void BigInt_destroy(BigInt& bi); // Free memory dynamically allocated
29
30 void BigInt_inc(BigInt& num); // increase by 1
31
32 BigInt BigInt_add(const BigInt& a, const BigInt& b); // a + b
33
34 //////////////////////////////////////
35
36 //The n'th Fibonacci number n(1)=1,n(2)=1,n(3)=2,n(4)=3,n(5)=5,...
37 BigInt fibo(unsigned int n);
38
39 //sqrt(x)
40 BigInt sqrt(const BigInt& x);
41
42 //////////////////////////////////////
43 //assistance function (of my own)
44
45 int no_leading_0_len(const char* s);
46
47 //////////////////////////////////////
48 //Operator Overloads:
49
50 int operator==(const BigInt& a, const BigInt& b);
51 BigInt operator+(const BigInt& a, const BigInt& b);

```

```
52
53 //assistance function (of my own)
54 int operator<(const BigInt& a, const BigInt& b);
55
56
57
58
59 #endif
```