

```

1  #define _CRTDBG_MAP_ALLOC
2  #include <stdlib.h>
3  #include <crtdbg.h>
4
5  #ifdef _DEBUG
6  #ifndef DBG_NEW
7  #define DBG_NEW new ( _NORMAL_BLOCK , __FILE__ , __LINE__ )
8  #define new DBG_NEW
9  #endif
10 #endif // _DEBUG
11 //-----
12 #include <iostream>
13 using namespace std;
14 #include "BigInt.h"
15 #include <string.h>
16 //-----
17
18 void BigInt_inc(BigInt& num)
19 {
20     int carry = 1; // initialize carry for 1 to increase in 1 at the start
21     for (int i = num.n_digits - 1; i >= 0; i--)
22     {
23         int temp = num.digits[i] - '0' + carry;
24         num.digits[i] = temp % 10 + '0';
25         carry = temp / 10;
26     }
27     if (carry)
28     {
29         char* temp = new char[num.n_digits + 1];
30         for (int i = 0; i < num.n_digits; i++)
31             temp[i + 1] = num.digits[i];
32         temp[0] = '0' + 1;
33         num.n_digits++;
34         delete[] num.digits;
35         num.digits = temp;
36     }
37 }
38 BigInt BigInt_add(const BigInt& a, const BigInt& b)
39 {
40     int carry = 0;
41     BigInt myint = { 0,0 };
42     int max, min, aindex, bindex; // aindex is index off a and bindex is
43     // the same with b
44     bindex = b.n_digits - 1;
45     aindex = a.n_digits - 1;
46     if (a.n_digits >= b.n_digits)
47     {
48         max = a.n_digits;
49         min = b.n_digits;
50     }
51     else
52     {
53         max = b.n_digits;

```

```
53     min = a.n_digits;
54 }
55 myint.digits = new char[max];
56 myint.n_digits = max;
57 for (int i = max - 1; i >= 0; i--)
58 {
59     if (aindex < 0)
60     {
61         myint.digits[i] = (b.digits[bindex] - '0' + carry) % 10 + '0';
62         carry = (b.digits[bindex] - '0' + carry) / 10;
63     }
64     else if (bindex < 0)
65     {
66         myint.digits[i] = (a.digits[aindex] - '0' + carry) % 10 + '0';
67         carry = (a.digits[aindex] - '0' + carry) / 10;
68     }
69     else
70     {
71         myint.digits[i] = (a.digits[aindex] - '0' + b.digits[bindex] - ➤
72             '0' + carry) % 10 + '0';
73         carry = (a.digits[aindex] - '0' + b.digits[bindex] - '0' + ➤
74             carry) / 10;
75     }
76     aindex--;
77     bindex--;
78 }
79 if (carry)
80 {
81     char* temp = new char[myint.n_digits + 1];
82     for (int i = 0; i < myint.n_digits; i++)
83         temp[i + 1] = myint.digits[i];
84     delete[] myint.digits;
85     myint.digits = temp;
86     myint.n_digits++;
87     myint.digits[0] = '0' + 1;
88 }
89 return myint;
90 }
```