

```

1  #include <iostream>
2  #include "HugeInteger.h"
3  using namespace std;
4
5  int main()
6  {
7      HugeInteger num1("63434");
8      HugeInteger num2("4534");
9      HugeInteger total;
10
11     num1.add(num2, total);
12     cout << "num1 + num2: ";
13     num1.output(total);
14     num1.subtract(num2, total);
15     cout << "\nThe difference between two integers: ";
16     num1.output(total);
17     num1.multiply(num2, total);
18     cout << "\nnum1 * num2: ";
19     num1.output(total);
20     //num1.divide(num2, total);
21     cout << "\n";
22     //num1.modulus(num2, total);
23
24     cout << "num1 is equal to num2: ";
25     num1.isEqualTo(num2);
26
27     cout << "num1 is not equal to num2: ";
28     num1.isNotEqualTo();
29
30     cout << "num1 is greater than num2: ";
31     num1.isGreaterThan();
32
33     cout << "num1 is less than num2: ";
34     num1.isLessThan();

```

```

36     cout << "num1 is greater than or equal to num2: ";
37     num1.isGreaterThanOrEqualTo();
38
39     cout << "num1 is less than or equal to num2: ";
40     num1.isLessThanOrEqualTo();
41
42     cout << "num1 is zero: ";
43     num1.isZero();
44
45     return 0;
46 }

```

```

1  #ifndef HugeInteger_H
2  #define HugeInteger_H
3  #include <string>
4  using namespace std;
5
6  class HugeInteger{
7  public:
8      static const int SIZE = 40;
9      int digits[SIZE] = {};
10
11      HugeInteger(string = "0");
12      void input(string);
13      void output(HugeInteger &);
14
15      void add(HugeInteger &, HugeInteger &);
16      void subtract(HugeInteger &, HugeInteger &);
17      void multiply(HugeInteger &, HugeInteger &);
18      void divide(HugeInteger &, HugeInteger &);
19      void modulus(HugeInteger &, HugeInteger &);
20
21      void isEqualTo(HugeInteger &);
22      void isNotEqualTo();
23      void isGreaterThan();
24      void isLessThan();
25      void isGreaterThanOrEqualTo();
26      void isLessThanOrEqualTo();
27      void isZero();
28
29  private:
30      string inputValue;
31      int valueLength;
32      int largerLength;
33      int lengthAfterCalculation;
34      int largerInteger;
35      int equalCounter = 0;
36      int zeroCounter = 0;
37
38      void beforeSubtract(HugeInteger &);
39  };
40
41 #endif

```

```

1  #include <iostream>
2  #include <string>
3  #include "HugeInteger.h"
4  using namespace std;
5
6  HugeInteger::HugeInteger(string inputValue){
7      input(inputValue);
8  }
9
10 void HugeInteger::input(string inputValue){
11
12     valueLength = inputValue.length();
13
14     for (int i=39, j=inputValue.length()-1; i>=0, j>=0; i--, j--){
15         digits[i] = inputValue[j] - '0';
16     }
17 }
18
19
20 void HugeInteger::output(HugeInteger &total){
21     if (lengthAfterCalculation != 0){
22         for (int i=40-lengthAfterCalculation; i<40; i++){
23             cout << total.digits[i];
24         }
25     }
26     else if (lengthAfterCalculation == 0){
27         cout << 0;
28     }
29     lengthAfterCalculation = 0;
30     cout << endl;
31 }
32
33

```

```

34 void HugeInteger::add(HugeInteger &num2, HugeInteger &total){
35
36     // 清空儲存格
37     for (int i=0; i<40; i++){
38         total.digits[i] = 0;
39     }
40
41     // 判斷哪個integer的位數比較大
42     if (valueLength >= num2.valueLength){
43         largerLength = valueLength;
44     }
45     else{
46         largerLength = num2.valueLength;
47     }
48
49     // 相加
50     for (int i=39; i>=39-largerLength; i--){
51         total.digits[i] += digits[i] + num2.digits[i];
52     }
53
54     // 進位處理
55     for (int i=39; i>=0; i--){
56         if (total.digits[i] >= 10){
57             total.digits[i] = total.digits[i] % 10;
58             total.digits[i-1] += 1;
59         }
60     }
61
62     // 算相加後的位數
63     for (int i=0; i<40; i++){
64         if (total.digits[i] > 0){
65             lengthAfterCalculation = 40 - i;
66             break;
67         }
68     }
69 }
70 }
71

```

```

72 void HugeInteger::beforeSubtract(HugeInteger &num2){
73     if (valueLength > num2.valueLength){
74         largerInteger = 1;
75     }
76     else if (valueLength == num2.valueLength and digits[40-valueLength] > num2.digits[40-num2.valueLength]){
77         largerInteger = 1;
78     }
79     else if (valueLength < num2.valueLength){
80         largerInteger = 2;
81     }
82     else if (valueLength == num2.valueLength and digits[40-valueLength] < num2.digits[40-num2.valueLength]){
83         largerInteger = 2;
84     }
85     else if (valueLength == num2.valueLength and digits[40-valueLength] == num2.digits[40-num2.valueLength]){
86         largerInteger = 0;
87     }
88 }
89 }
90

```

```

90
91 void HugeInteger::subtract(HugeInteger &num2, HugeInteger &total){
92
93     // 清空儲存格
94     for (int i=0; i<40; i++){
95         total.digits[i] = 0;
96     }
97
98     // 相減且做借位處理
99     beforeSubtract(num2);
100     // num1 大
101     if (largerInteger == 1 or largerInteger == 0){
102
103         int temp[40];
104         for (int i=0; i<40; i++){
105             temp[i] = digits[i];
106         }
107
108         for (int i=39; i>=39-largerLength; i--){
109             if (temp[i] - num2.digits[i] < 0){
110                 total.digits[i] = temp[i] + 10 - num2.digits[i];
111                 temp[i-1] -= 1;
112             }
113             else if (temp[i] - num2.digits[i] >= 0){
114                 total.digits[i] = temp[i] - num2.digits[i];
115             }
116         }
117     }
118
119     // num2 大
120     else if (largerInteger == 2){
121
122         int temp[40];
123         for (int i=0; i<40; i++){
124             temp[i] = num2.digits[i];
125         }
126
127         for (int i=39; i>=39-largerLength; i--){
128             if (temp[i] - digits[i] < 0){
129                 total.digits[i] = temp[i] + 10 - digits[i];
130                 temp[i-1] -= 1;
131             }
132             else if (temp[i] - digits[i] >= 0){
133                 total.digits[i] = temp[i] - digits[i];
134             }
135         }
136     }
137 }
138

```

```

139
140 // 算相減後的位數
141 for (int i=0; i<40; i++){
142     if (total.digits[i] > 0){
143         lengthAfterCalculation = 40 - i;
144         break;
145     }
146 }
147
148
149 void HugeInteger::multiply(HugeInteger &num2, HugeInteger &total){
150
151     // 清空儲存格
152     for (int i=0; i<40; i++){
153         total.digits[i] = 0;
154     }
155
156     // num1 * num2
157     int k = 39;
158     for (int i=39; i>=40-num2.valueLength; i--){
159         k = i;
160         for (int j=39; j>=40-valueLength; j--){
161             total.digits[k] += digits[j] * num2.digits[i];
162
163             // 進位處理
164             if (total.digits[k] >= 10){
165                 total.digits[k-1] += total.digits[k] / 10;
166                 total.digits[k] %= 10;
167             }
168             k--;
169         }
170     }
171
172
173     // 算相乘後的位數
174     for (int i=0; i<40; i++){
175         if (total.digits[i] > 0){
176             lengthAfterCalculation = 40 - i;
177             break;
178         }
179     }
180 }
181
182

```

```

183 - void HugeInteger::divide(HugeInteger &num2, HugeInteger &total){
184
185     // 清空儲存格
186     for (int i=0; i<40; i++){
187         total.digits[i] = 0;
188     }
189
190 }
191
192
193 - void HugeInteger::modulus(HugeInteger &num2, HugeInteger &total){
194
195     // 清空儲存格
196     for (int i=0; i<40; i++){
197         total.digits[i] = 0;
198     }
199
200 }
201
202 }
203
204 - void HugeInteger::isEqualTo(HugeInteger &num2){
205     for (int i=0; i<40; i++){
206         if (digits[i] == num2.digits[i]){
207             equalCounter++;
208             if (equalCounter == 40){
209                 cout << "True" << endl;
210                 break;
211             }
212         }
213         else if (digits[i] != num2.digits[i]){
214             cout << "False" << endl;
215             break;
216         }
217     }
218 }
219
220
221 - void HugeInteger::isNotEqualTo(){
222     if (equalCounter == 40){
223         cout << "False" << endl;
224     }
225     else{
226         cout << "True" << endl;
227     }
228 }
229 }
230

```

```

231 - void HugeInteger::isGreaterThan(){
232     if (largerInteger == 1){
233         cout << "True" << endl;
234     }
235     else{
236         cout << "False" << endl;
237     }
238 }
239 }
240
241 - void HugeInteger::isLessThan(){
242     if (largerInteger == 1 or largerInteger == 0){
243         cout << "False" << endl;
244     }
245     else{
246         cout << "True" << endl;
247     }
248 }
249 }
250
251 - void HugeInteger::isGreaterThanOrEqualTo(){
252     if (largerInteger == 1 or largerInteger == 0){
253         cout << "True" << endl;
254     }
255     else{
256         cout << "False" << endl;
257     }
258 }
259 }
260
261 - void HugeInteger::isLessThanOrEqualTo(){
262     if (largerInteger == 2 or largerInteger == 0){
263         cout << "True" << endl;
264     }
265     else{
266         cout << "False" << endl;
267     }
268 }
269 }
270

```

```

270
271 void HugeInteger::isZero(){
272     for (int i=0; i<40; i++){
273         if (digits[i] != 0){
274             cout << "False" << endl;
275             break;
276         }
277         zeroCounter++;
278     }
279     if (zeroCounter == 40){
280         cout << "True" << endl;
281     }
282 }
283
284

```

num1 + num2: 67968

The difference between two integers: 58900

num1 * num2: 287609756

num1 is equal to num2: False

num1 is not equal to num2: True

num1 is greater than num2: True

num1 is less than num2: False

num1 is greater than or equal to num2: True

num1 is less than or equal to num2: False

num1 is zero: False