高精度加减法

无论是32位还是64位整数，都有大小的范围限制。高精度算法主要解决的就是超过C++内置整型的运算。在求大整数运算的时候，由于输入数字太大，我们需要以字符串形式读入；输出同理。我们假定输入里没有前导零（前导零的意思是，一个非零数的最高位的那些零，例如0030这个数的左起的两个零。这些零在最终答案中不应该出现）。

本文主要考虑高精度加减法。

高精度加法

模拟人工加法，人工做加法的时候，是由低位向高位，逐位相加；某一位相加超过10，就要考虑进位，下一位要把进位加上。例题：https://blog.csdn.net/qq_46105170/article/details/113793376。看代码：

```
#include <algorithm>
#include <iostream>
#include <string>

using namespace std;

string add(auto& a, auto& b) {
  string s;
  s.reserve(max(a.size(), b.size()) + 1);

  // t存储进位
  for (int i = a.size() - 1, j = b.size() - 1, t = 0; i >= 0 || j >= 0 || t;) {
    int sum = t;
    if (i >= 0) sum += a[i--] - '0';
    if (j >= 0) sum += b[j--] - '0';
    s.push_back(sum % 10 + '0');
    t = sum / 10;
  }

  reverse(s.begin(), s.end());
  return s;
}

int main() {
  string a, b;
  getline(cin, a);
  getline(cin, b);

```

```
  cout << add(a, b) << endl;
}
```

高进度减法

一样，也要模拟人工做减法的流程。但与高精度加法不同的是，人工做减法的时候，我们总是用大数减去小数。所以在读入输入a和b之后，首先要判断大小。如果$a < b$，那么我们要求的是$b - a$，当然最后答案之前也要加上负号。此外还需要注意：

1. 与加法类似，做减法的时候也需要从最低位做起
2. 当前位做减法的时候要注意考虑之前的退位，当前位做减法的时候如果变为负数，也要记录退位，在做下一位减法的时候考虑进去
3. 最后答案可能会出现前导零，要注意去掉。

例题：https://blog.csdn.net/qq_46105170/article/details/113793404。

```cpp
#include <algorithm>
#include <iostream>
#include <string>
using namespace std;

// 比较a和b这两个数哪个大。注意这里已经假定输入数里没有前导零了
bool cmp(auto& a, auto& b) {
  if (a.size() != b.size()) return a.size() >= b.size();
  return a >= b;
}

string sub(auto& a, auto& b) {
  string c;
  c.reserve(a.size());

  // 从最低位开始做减法。t是退位
  for (int i = 0, t = 0; i < a.size(); ++i) {
    int x = a[a.size() - 1 - i] - '0';
    int y = (i < b.size() ? b[b.size() - 1 - i] - '0' : 0);
    // 当前位做减法要把退位考虑进去
    int diff = x - y - t;

    if (diff < 0) {
      diff += 10;
      t = 1;
    } else
      t = 0;

    c += diff + '0';
```

```
  }
  // 去掉前导零
  while (c.size() > 1 && c.back() == '0') c.pop_back();
  // c是逆序的答案，还要翻转回来
  reverse(c.begin(), c.end());
  return c;
}

int main() {
  string a, b;
  getline(cin, a);
  getline(cin, b);

  cout << (cmp(a, b) ? sub(a, b) : "-" + sub(b, a)) << endl;
}
```

作业

上面两道例题