if (plus != 0)

cout << plus;

if (a.size() - a\_point > b.size() - b\_point) {

for (int i = b.size() - b\_point; i < a.size() - a\_point; i++) {

if (i != 0) {

single = a[a\_point + i];

singles.push\_back(single);

}

}

}

if (a.size() - a\_point < b.size() - b\_point) {

for (int i = a.size() - a\_point; i < b.size() - b\_point; i++) {

if(i!=0) {

single = b[b\_point + i];

singles.push\_back(single);

}

}

}

if ((a.size() == a\_point) && (b.size() == b\_point)) {

single = '0';

singles.push\_back(single);

}

int j = 0;

for (int i = max(a.size() - a\_point, b.size() - b\_point); i > 0; i--) {

if (singles[i] != 0) {

for (int i = singles.size() - 1; i >= 0; i--) {

cout << singles[i];

if (i == singles.size() - number\_after - 1)

cout << '.';

}

j = 7;

}

}

if (j==7){

for (int i = singles.size() ; i >= singles.size() - number\_after; i--) {

cout << singles[i];

cout << '.0';

}

}

// ConsoleApplication11.cpp : 定义控制台应用程序的入口点。

//

#include <iostream>

#include<string>

#include<algorithm>

#include <vector>

using namespace std;

class input\_error {};

string calculator(string a, string b) {

int a\_point, b\_point;

int h = 0;

cout << a.find('.', 0);

if (a.find('.', 0) <a.size())

a\_point = a.find(".", 0);

else

a\_point = a.size();

cout<< b.find('.', 0) << endl;

if (b.find('.', 0) <b.size())

b\_point = b.find('.', 0);

else

b\_point = b.size();

cout << a\_point << b\_point<<endl;

vector<int>singles;

int plus = 0;

int single = 0;

int number\_before, number\_after;

if (a\_point > b\_point)

number\_before = b\_point;

else number\_before = a\_point;

if (a.size() - a\_point > b.size() - b\_point) {

if (b.size() != b\_point)

number\_after = b.size() - b\_point - 1;

else number\_after = 0;

}

else {

if(a.size()!=a\_point)

number\_after = a.size() - a\_point - 1;

else number\_after = 0;

}

cout << number\_before << number\_after << endl;

for (int i = number\_after; i >= -number\_before; i--) {

cout << i << endl;

if (i!=0) {

plus += (a[a\_point + i] - '0') + (b[b\_point + i] - '0');

cout << a[a\_point + i] - '0'<<endl << b[b\_point + i] - '0' << endl;

single = plus % 10;

plus = plus / 10;

singles.push\_back(single);

cout << "plus:" << plus<<endl;

cout << "single:" << single<<endl;

}

}

cout << "plus:" << plus<<endl;

cout << "single:" << single << endl;

single = 0;

if (a\_point > b\_point) {

for (int i = a\_point - b\_point - 1; i >= 0; i--) {

single = plus + (a[i] - '0');

plus = single / 10;

single %= 10;

cout << i << endl;

cout << "a[i]:" << a[i] << endl;

cout << "single:" << single << endl;

cout << "plus:" << plus << endl;

singles.push\_back(single);

}

}

else {

for (int i = b\_point - a\_point - 1; i >= 0; i--) {

single = plus + (b[i] - '0');

plus = single / 10;

single %= 10;

cout << i << endl;

cout << "b[i]:" << b[i] << endl;

cout << "single:" << single << endl;

cout << "plus:" << plus << endl;

singles.push\_back(single);

}

}

if (plus != 0)

cout << plus;

vector<int>befores;

int before = 0;

if (a.size() - a\_point > b.size() - b\_point) {

for (int i = a.size() - b\_point; i < b.size() - a\_point; i--) {

if (i != 0) {

before = a[a\_point + i];

befores.push\_back(before);

}

}

}

if (a.size() - a\_point < b.size() - b\_point) {

for (int i = b.size() - a\_point; i < a.size() - b\_point; i--) {

if (i != 0) {

before = b[b\_point + i];

befores.push\_back(before);

}

}

}

if ((a.size() == a\_point) && (b.size() == b\_point)) {

before = '0';

befores.push\_back(before);

}

for (int i = singles.size() - 1; i >= 0; i--) {

before = singles[i];

befores.push\_back(before);

}

int j = 0;

for (int i = max(a.size() - a\_point, b.size() - b\_point); i > 0; i--) {

if (befores[i] != 0) {

for (int i = befores.size() - 1; i >= 0; i--) {

cout << befores[i];

if (i == befores.size() - number\_after - 1)

cout << '.';

}

j = 7;

}

}

if (j == 7) {

for (int i = befores.size(); i >= befores.size() - number\_after; i--) {

cout << befores[i];

cout << '.0';

}

}

return 0;

}

bool judgement(string a) {

int j = 0;

for (int i = 0; i < a.size(); i++) {

if (a[i] == '.')

j += 1;

if (j > 1)

return 0;

}

return 1;

}

int main()

{

string a, b,operation;

bool a\_bool, b\_bool;

cout << "input:";

cin >> operation >> a >> b;

for (unsigned int i = 0; i < a.length(); i++) {

a\_bool = ((('0' <= a[i]) && (a[i]<= '9'))||(a[i] == '.')) && judgement(a);

}

for (unsigned int i = 0; i < b.length(); i++) {

b\_bool = ((('0' <= b[i]) && (b[i]<= '9'))|| (b[i]== '.')) && judgement(b);

}

cout << a\_bool << ' ' << b\_bool << endl;

try {

if (!(a\_bool&& b\_bool && (operation == "+")))

throw input\_error{};

calculator(a, b);

}

catch (input\_error) {

cerr << "input error";

}

return 0;

}

if ((a.size() == a\_point) && (b.size() == b\_point)) {

before = 0;

befores.push\_back(before);

cout << "before:" << before <<"a.size() == a\_point) && (b.size() == b\_point"<< endl;

#include <iostream>

#include<string>

#include<algorithm>

#include <vector>

using namespace std;

class input\_error {};

string calculator(string a, string b) {

int a\_point, b\_point;

if (a.find('.', 0) <a.size())

a\_point = a.find(".", 0);

else

a\_point = a.size();

if (b.find('.', 0) <b.size())

b\_point = b.find('.', 0);

else

b\_point = b.size();

vector<int>singles;

int plus = 0;

int single = 0;

int number\_before, number\_after;

if (a\_point > b\_point)

number\_before = b\_point;

else number\_before = a\_point;

if (a.size() - a\_point > b.size() - b\_point) {

if (b.size() != b\_point)

number\_after = b.size() - b\_point - 1;

else number\_after = 0;

}

else {

if(a.size()!=a\_point)

number\_after = a.size() - a\_point - 1;

else number\_after = 0;

}

for (int i = number\_after; i >= -number\_before; i--) {

if (i != 0) {

plus += (a[a\_point + i] - '0') + (b[b\_point + i] - '0');

single = plus % 10;

plus = plus / 10;

singles.push\_back(single);

}

}

single = 0;

if (a\_point > b\_point) {

for (int i = a\_point - b\_point - 1; i >= 0; i--) {

single = plus + (a[i] - '0');

plus = single / 10;

single %= 10;

singles.push\_back(single);

}

}

else {

for (int i = b\_point - a\_point - 1; i >= 0; i--) {

single = plus + (b[i] - '0');

plus = single / 10;

single %= 10;

singles.push\_back(single);

}

}

if (plus != 0)

cout << plus;

vector<int>befores;

int before = 0;

for (int i = singles.size() - 1; i >= 0; i--) {

before = singles[i];

befores.push\_back(before);

}

if (a.size() - a\_point > b.size() - b\_point) {

for (int i = a.size() - a\_point-1; i > b.size() - b\_point-1; i--) {

if (i != 0) {

before = a[a\_point + i]-'0';

befores.push\_back(before);

}

}

}

if (a.size() - a\_point < b.size() - b\_point) {

for (int i = b.size() - b\_point-1; i > a.size() - a\_point-1; i--) {

if (i !=0) {

before = b[b\_point + i]-'0';

befores.push\_back(before);

}

}

}

int j = 0;

for (int i = befores.size(); i >(befores.size()-number\_after); i--) {

if (befores[i-1] != 0) {

j = 7;

}

}

if (j == 7) {

for (int i = 0; i < befores.size(); i++) {

cout << befores[i];

if (i == befores.size() - max(a.size() - a\_point, b.size() - b\_point))

cout << '.';

}

}

if (j == 0) {

for (int i = 0; i < max(a\_point,b\_point); i++) {

cout << befores[i];

}

cout << ".0";

}

return 0;

}

bool judgement(string a) {

int j = 0;

for (int i = 0; i < a.size(); i++) {

if (a[i] == '.')

j += 1;

if (j > 1)

return 0;

}

return 1;

}

int main()

{

string a, b,operation;

bool a\_bool, b\_bool;

cout << "input:";

cin >> operation >> a >> b;

for (unsigned int i = 0; i < a.length(); i++) {

a\_bool = ((('0' <= a[i]) && (a[i]<= '9'))||(a[i] == '.')) && judgement(a);

}

for (unsigned int i = 0; i < b.length(); i++) {

b\_bool = ((('0' <= b[i]) && (b[i]<= '9'))|| (b[i]== '.')) && judgement(b);

}

try {

if (!(a\_bool&& b\_bool && (operation == "+")))

throw input\_error{};

calculator(a, b);

}

catch (input\_error) {

cerr << "input error";

}

return 0;

}

#include <iostream>

#include<string>

#include<algorithm>

#include <vector>

using namespace std;

class input\_error {};

string calculator(string a, string b) {

int a\_point, b\_point;

if (a.find('.', 0) <a.size())

a\_point = a.find(".", 0);

else

a\_point = a.size();

if (b.find('.', 0) <b.size())

b\_point = b.find('.', 0);

else

b\_point = b.size();

vector<int>singles;

int plus = 0;

int single = 0;

int number\_before, number\_after;

if (a\_point > b\_point)

number\_before = b\_point;

else number\_before = a\_point;

if (a.size() - a\_point > b.size() - b\_point) {

if (b.size() != b\_point)

number\_after = b.size() - b\_point - 1;

else number\_after = 0;

}

else {

if(a.size()!=a\_point)

number\_after = a.size() - a\_point - 1;

else number\_after = 0;

}

for (int i = number\_after; i >= -number\_before; i--) {

if (i != 0) {

plus += (a[a\_point + i] - '0') + (b[b\_point + i] - '0');

single = plus % 10;

plus = plus / 10;

singles.push\_back(single);

}

}

single = 0;

if (a\_point > b\_point) {

for (int i = a\_point - b\_point - 1; i >= 0; i--) {

single = plus + (a[i] - '0');

plus = single / 10;

single %= 10;

singles.push\_back(single);

}

}

else {

for (int i = b\_point - a\_point - 1; i >= 0; i--) {

single = plus + (b[i] - '0');

plus = single / 10;

single %= 10;

singles.push\_back(single);

}

}

if (plus != 0)

cout << plus;

vector<int>befores;

int before = 0;

for (int i = singles.size() - 1; i >= 0; i--) {

before = singles[i];

befores.push\_back(before);

//cout << "before:" << before << endl;

}

if (a.size() - a\_point > b.size() - b\_point) {

for (int i = b.size() - b\_point; i <a.size() - a\_point; i++) {

if (i != 0) {

before = a[a\_point + i]-'0';

befores.push\_back(before);

//cout << "before:" << before << endl;

}

}

}

if (a.size() - a\_point < b.size() - b\_point) {

for (int i = a.size() - a\_point; i < b.size() - b\_point; i++) {

if (i !=0) {

before = b[b\_point + i]-'0';

befores.push\_back(before);

//cout << "before:" << before << endl;

}

}

}

int j = 0;

for (int i = befores.size(); i >(befores.size()-number\_after); i--) {

if (befores[i-1] != 0) {

j = 7;

// cout << "before:" << before << endl;

}

}

//cout << ' ' << "j:" << j << endl;

if (j == 7) {

for (int i = 0; i < befores.size(); i++) {

cout << befores[i];

if (i == befores.size() - max(a.size() - a\_point, b.size() - b\_point))

cout << '.';

}

}

if (j == 0) {

for (int i = 0; i < max(a\_point,b\_point); i++) {

cout << befores[i];

}

cout << ".0";

}

return 0;

}

bool judgement(string a) {

int j = 0;

for (int i = 0; i < a.size(); i++) {

if (a[i] == '.')

j += 1;

if (j > 1)

return 0;

}

return 1;

}

int main()

{

string a, b,operation;

bool a\_bool, b\_bool;

cout << "input:";

cin >> operation >> a >> b;

for (unsigned int i = 0; i < a.length(); i++) {

a\_bool = ((('0' <= a[i]) && (a[i]<= '9'))||(a[i] == '.')) && judgement(a);

}

for (unsigned int i = 0; i < b.length(); i++) {

b\_bool = ((('0' <= b[i]) && (b[i]<= '9'))|| (b[i]== '.')) && judgement(b);

}

try {

if (!(a\_bool&& b\_bool && (operation == "+")))

throw input\_error{};

calculator(a, b);

}

catch (input\_error) {

cerr << "input error";

}

return 0;

}

#include <iostream>

#include<string>

#include<algorithm>

#include <vector>

using namespace std;

class input\_error {};

string calculator(string a, string b) {

int a\_point, b\_point;

if (a.find('.', 0) <a.size())

a\_point = a.find(".", 0);

else

a\_point = a.size();

if (b.find('.', 0) <b.size())

b\_point = b.find('.', 0);

else

b\_point = b.size();

vector<int>singles;

int plus = 0;

int single = 0;

int number\_before, number\_after;

if (a\_point > b\_point)

number\_before = b\_point;

else number\_before = a\_point;

if (a.size() - a\_point > b.size() - b\_point) {

if (b.size() != b\_point)

number\_after = b.size() - b\_point - 1;

else number\_after = 0;

}

else {

if(a.size()!=a\_point)

number\_after = a.size() - a\_point - 1;

else number\_after = 0;

}

for (int i = number\_after; i >= -number\_before; i--) {

if (i != 0) {

plus += (a[a\_point + i] - '0') + (b[b\_point + i] - '0');

single = plus % 10;

plus = plus / 10;

singles.push\_back(single);

}

}

single = 0;

if (a\_point > b\_point) {

for (int i = a\_point - b\_point - 1; i >= 0; i--) {

single = plus + (a[i] - '0');

plus = single / 10;

single %= 10;

singles.push\_back(single);

}

}

else {

for (int i = b\_point - a\_point - 1; i >= 0; i--) {

single = plus + (b[i] - '0');

plus = single / 10;

single %= 10;

singles.push\_back(single);

}

}

int x = max(a\_point, b\_point);

if (plus != 0) {

cout << plus;

x++;

}

vector<int>befores;

int before = 0;

for (int i = singles.size() - 1; i >= 0; i--) {

before = singles[i];

befores.push\_back(before);

//cout << "before:" << before << endl;

}

if (a.size() - a\_point > b.size() - b\_point) {

for (int i = b.size() - b\_point; i <a.size() - a\_point; i++) {

if (i != 0) {

before = a[a\_point + i]-'0';

befores.push\_back(before);

//cout << "before:" << before << endl;

}

}

}

if (a.size() - a\_point < b.size() - b\_point) {

for (int i = a.size() - a\_point; i < b.size() - b\_point; i++) {

if (i !=0) {

before = b[b\_point + i]-'0';

befores.push\_back(before);

//cout << "before:" << before << endl;

}

}

}

int j = 0;

//cout << befores.size() << ' ' << befores.size() - number\_after << endl;

for (int i = befores.size(); i >x; i--) {

if (befores[i-1] != 0) {

j = 7;

// cout << "before:" << before << endl;

}

}

//cout << ' ' << "j:" << j << endl;

if (j == 7) {

for (int i = 0; i < befores.size(); i++) {

cout << befores[i];

if (i == befores.size() - max(a.size() - a\_point, b.size() - b\_point))

cout << '.';

}

}

if (j == 0) {

for (int i = 0; i < max(a\_point,b\_point); i++) {

cout << befores[i];

}

cout << ".0";

}

return 0;

}

bool judgement(string a) {

int j = 0;

for (int i = 0; i < a.size(); i++) {

if (a[i] == '.')

j += 1;

if (j > 1)

return 0;

}

return 1;

}

int main()

{

string a, b,operation;

bool a\_bool, b\_bool;

cout << "input:";

cin >> operation >> a >> b;

for (unsigned int i = 0; i < a.length(); i++) {

a\_bool = ((('0' <= a[i]) && (a[i]<= '9'))||(a[i] == '.')) && judgement(a);

}

for (unsigned int i = 0; i < b.length(); i++) {

b\_bool = ((('0' <= b[i]) && (b[i]<= '9'))|| (b[i]== '.')) && judgement(b);

}

try {

if (!(a\_bool&& b\_bool && (operation == "+")))

throw input\_error{};

calculator(a, b);

}

catch (input\_error) {

cerr << "input error";

}

return 0;

}

【100-1】