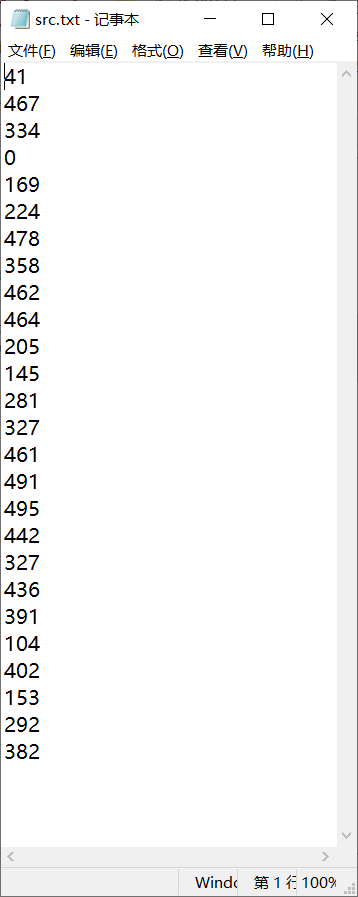
源文件(src.txt)：



1.

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

int main()

{

const char src[] = "src.txt";

int tmp, count = 0;

ifstream fin(src, ios::in);

if (!fin)

{

cerr << "Cannot open the file: " << src << endl;

return 1;

}

while (fin >> tmp)

{

cout << setw(8) << tmp << " ";

if ((++count) % 10 == 0) cout << endl;

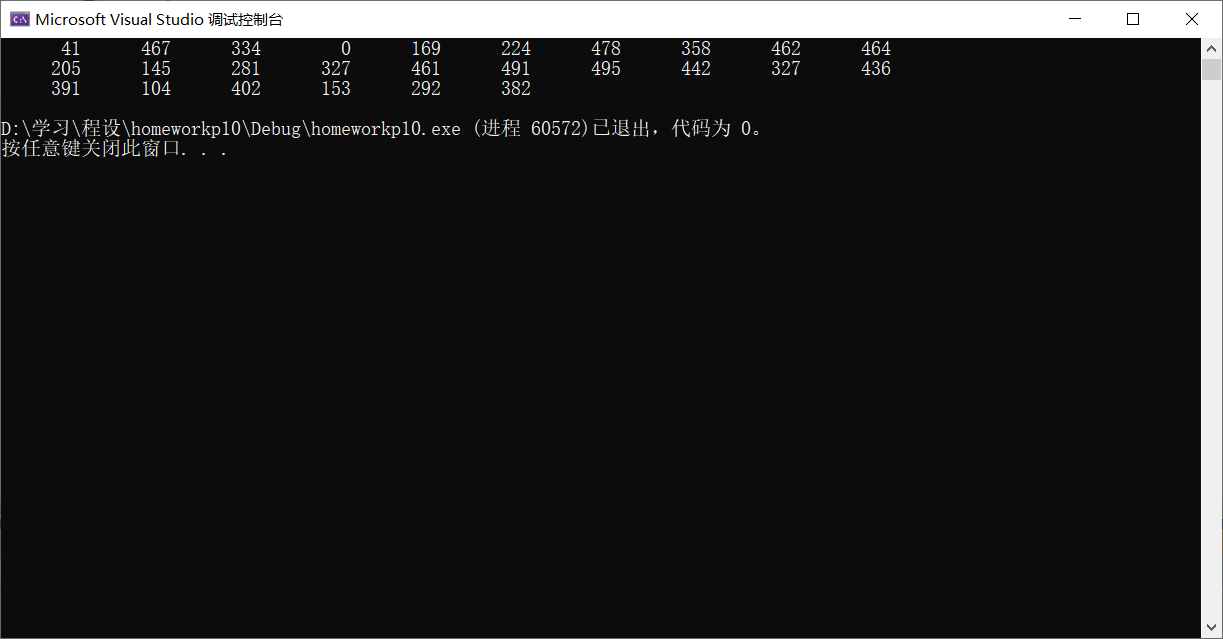
}

cout << endl;

fin.close();

return 0;

}



2.

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

int main()

{

const char src[] = "src.txt";

const char dest[] = "dest.dat";

int tmp;

ifstream fin(src, ios::in);

ofstream fout(dest, ios::out | ios::binary);

if (!fin)

{

cerr << "Cannot open the file: " << src;

return 1;

}

if (!fout)

{

cerr << "Cannot open the file: " << dest;

return 2;

}

while (fin >> tmp)

{

fout.write((const char\*)&tmp, sizeof(int));

}

fout.close();

fin.close();

fin.open(dest, ios::in | ios::binary);

if (!fin)

{

cerr << "Cannot open the file: " << dest;

return 3;

}

fin.seekg(5 \* sizeof(int), ios::beg);

int count = 0;

cout << left; //设置左对齐

while (true)

{

if (fin.read((char\*)&tmp, sizeof(int)))

{

cout << setw(8) << tmp << " ";

}

else break;

if (fin.read((char\*)&tmp, sizeof(int)))

{

cout << setw(8) << tmp << " ";

if ((++count) % 5 == 0) cout << endl;

}

else break;

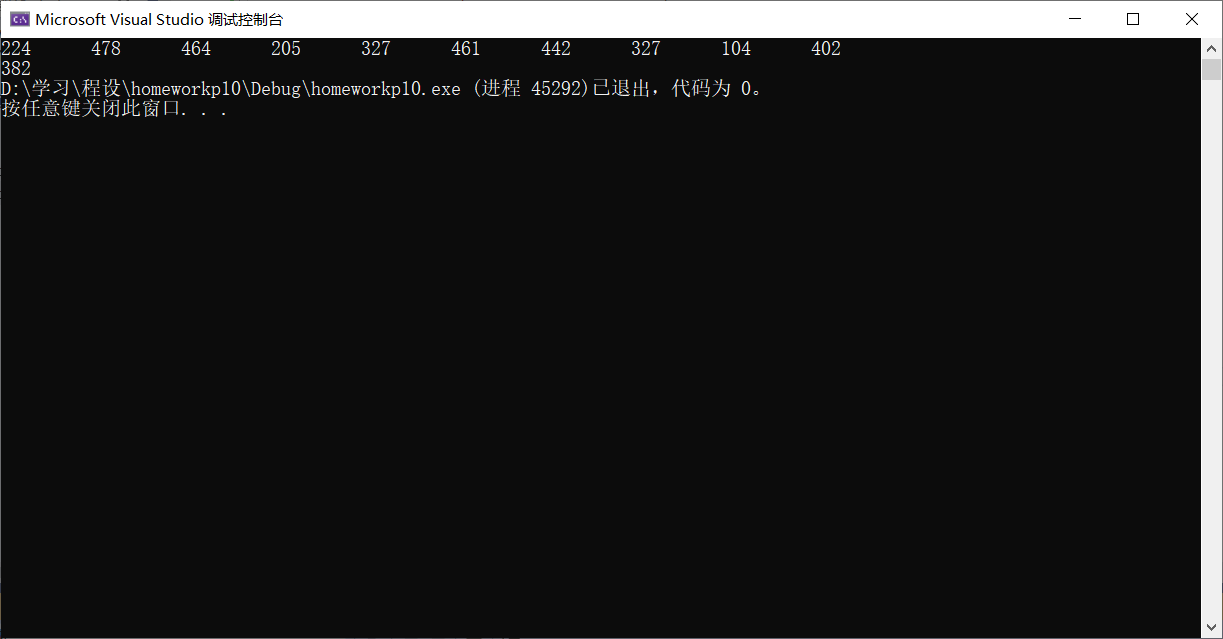
fin.seekg(2 \* sizeof(int), ios::cur);

}

fin.close();

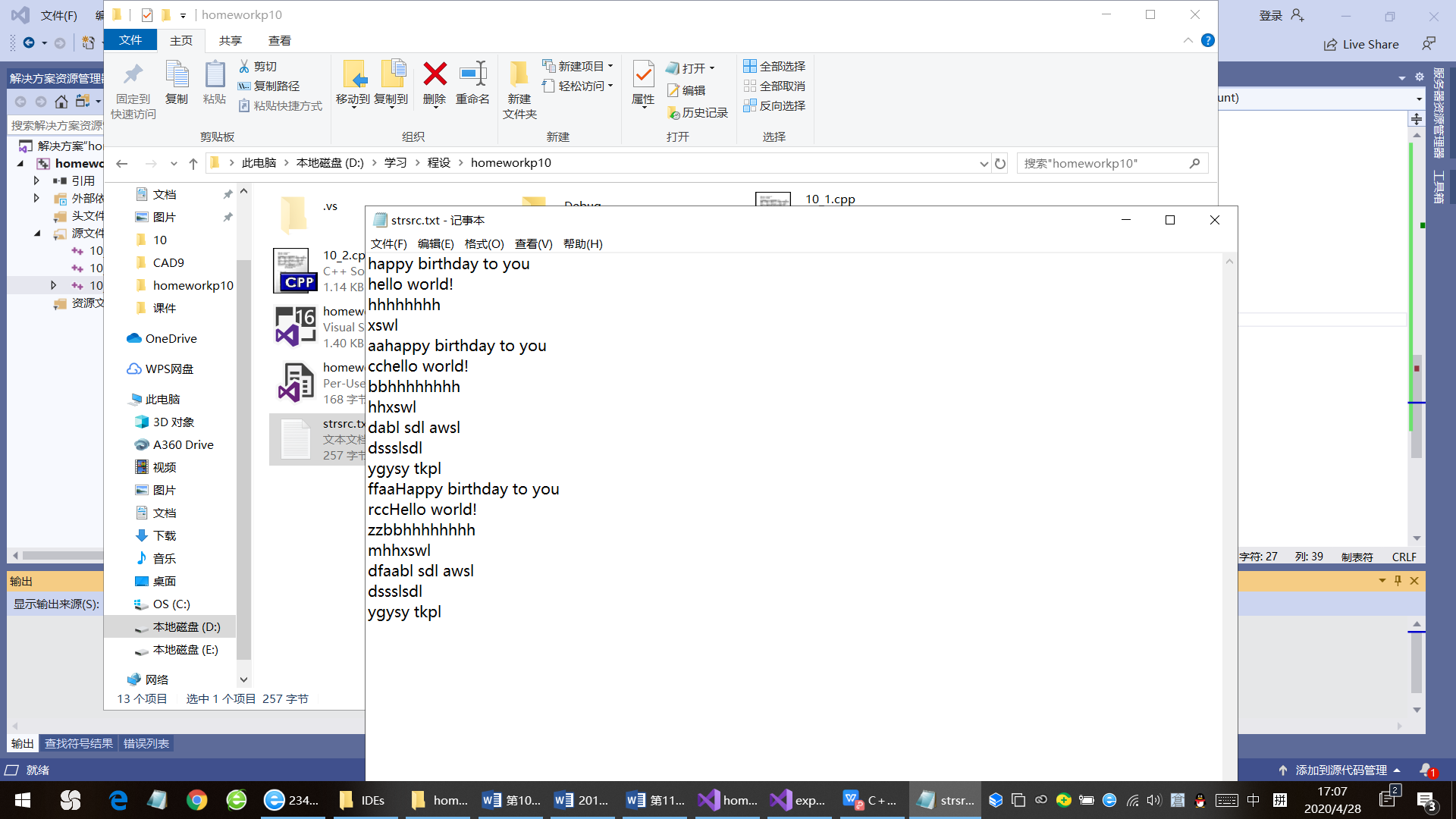
return 0;

}



**\*3.**

字符串源文件：



#include <iostream>

#include <fstream>

#include <cstring>

#pragma warning(disable:4996)

using namespace std;

void Sort(char\*\* strs, const int count);

int main()

{

int size = 4, count = 0;

char\*\* strs = new char\* [size];

const char src[] = "strsrc.txt";

const char dest[] = "strdest.txt";

for (int i = 0; i < size; ++i)

{

strs[i] = new char[85];

}

ifstream fin(src, ios::in);

if (!fin)

{

cerr << "Cannot open the file: " << src << endl;

return 1;

}

while (fin.getline(strs[count++], 81, '\n'))

{

if (count == size)

{

char\*\* newStrs = new char\* [size \*= 2];

for (int i = 0; i < count; ++i)

{

newStrs[i] = new char[81];

strcpy(newStrs[i], strs[i]);

delete[] strs[i];

}

for (int j = count; j < size; ++j)

newStrs[j] = new char[81];

delete[] strs;

strs = newStrs;

}

}

--count;

Sort(strs, count);

fin.close();

ofstream fout(dest, ios::out);

if (!fout)

{

for (int i = 0; i < size; ++i) delete[] strs[i];

delete[] strs;

cerr << "Cannot open the file: " << dest << endl;

return 2;

}

for (int i = 0; i < count; ++i)

{

fout << strs[i] << endl;

}

fout.close();

for (int i = 0; i < size; ++i) delete[] strs[i];

delete[] strs;

return 0;

}

void Sort(char\*\* strs, const int count)

{

int right = count - 1, now = 0;

while (right)

{

now = 0;

for (int i = 0; i < right; ++i)

{

if (strcmp(strs[i], strs[i + 1]) > 0)

{

now = i;

char\* tmp = strs[i];

strs[i] = strs[i + 1];

strs[i + 1] = tmp;

}

}

right = now;

}

}

结果：

