1.

#include <iostream>

#include <cstdlib>

using namespace std;

#define BAD\_NEW -1

#define BAD\_NEW\_TYPE int

#define OVERFLOW\_N -1.0

#define OVERFLOW\_TYPE double

#define UNDERFLOW\_N -1.0f

#define UNDERFLOW\_TYPE float

#define DELETE\_NULL\_PTR -1l

#define DELETE\_NULL\_PTR\_TYPE long

class Stack

{

private:

int\* p;

int N;

int max;

public:

Stack(int num)

{

p = new(std::nothrow) int[max = num];

if (p == NULL) throw BAD\_NEW;

N = 0;

}

void push\_back(int elem)

{

if (N == max) throw OVERFLOW\_N;

p[N++] = elem;

}

void pop\_back()

{

if (N == 0) throw UNDERFLOW\_N;

--N;

}

~Stack() { if (p == NULL) delete[] p; }

};

int main()

{

try

{

int a, b;

cin >> a;

cin >> b;

int\* p = new(std::nothrow) int[a];

if (p == NULL) throw BAD\_NEW;

delete[] p;

Stack s(5);

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

s.push\_back(i);

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

s.pop\_back();

int\* k = NULL;

if (k == NULL) throw DELETE\_NULL\_PTR;

else delete k;

}

catch (BAD\_NEW\_TYPE b)

{

cerr << "new failed!" << endl;

exit(1);

}

catch (OVERFLOW\_TYPE o)

{

cerr << "overflow!" << endl;

exit(1);

}

catch (UNDERFLOW\_TYPE u)

{

cerr << "underflow!" << endl;

exit(1);

}

catch (DELETE\_NULL\_PTR\_TYPE d)

{

cerr << "delete null pointer!" << endl;

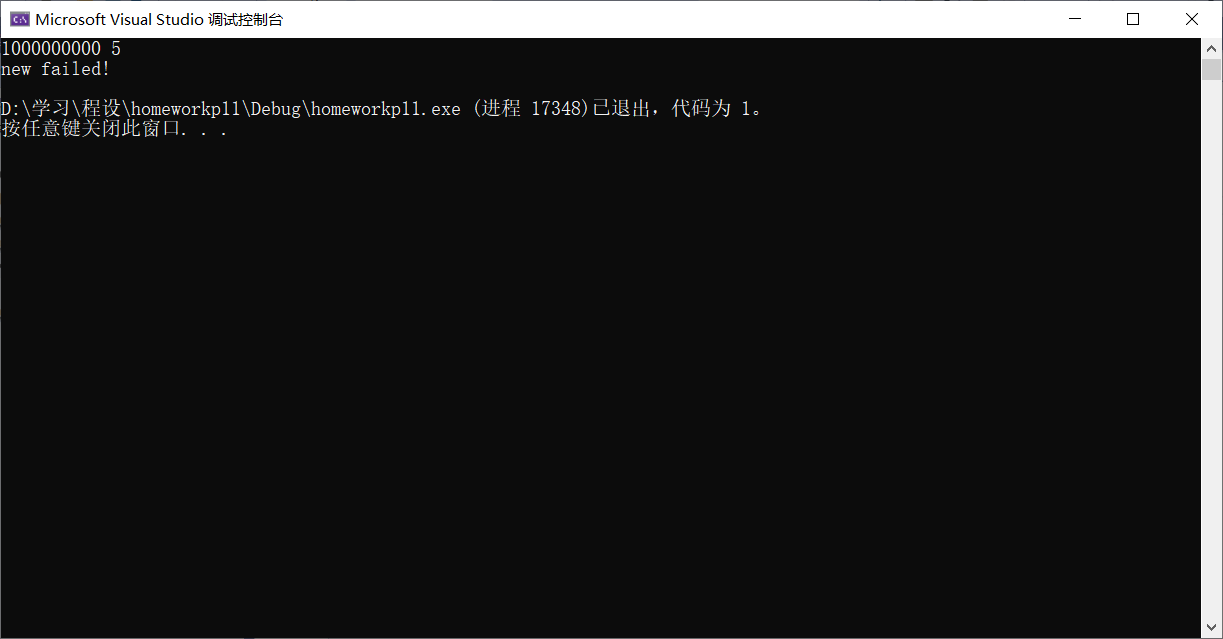
exit(1);

}

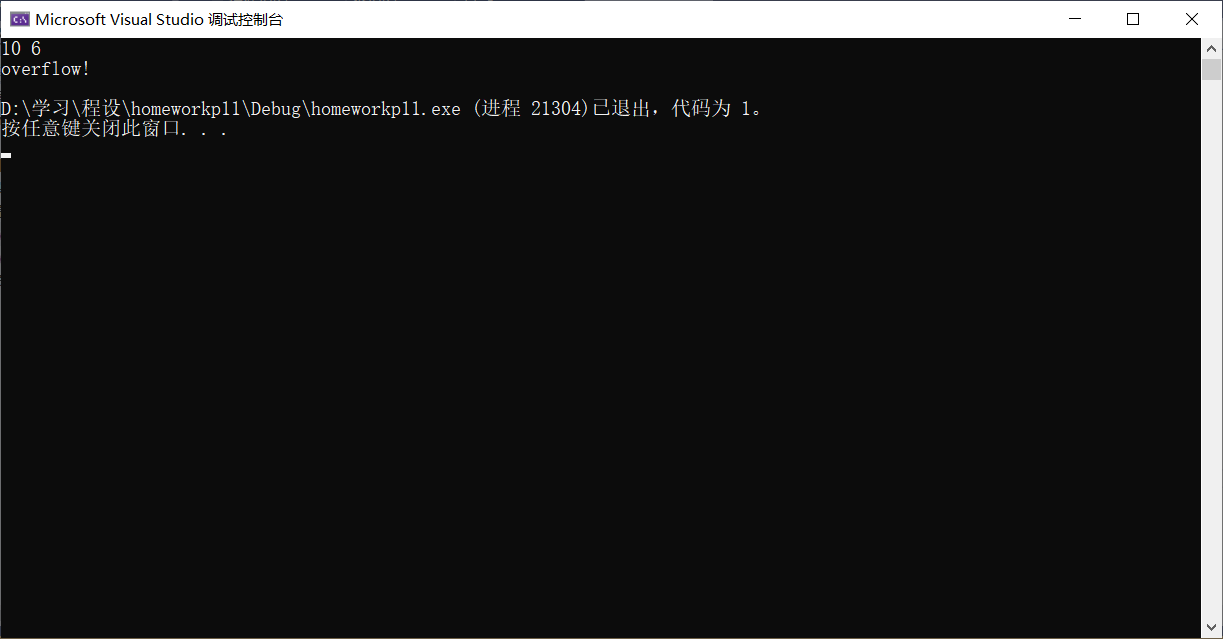
return 0;

}

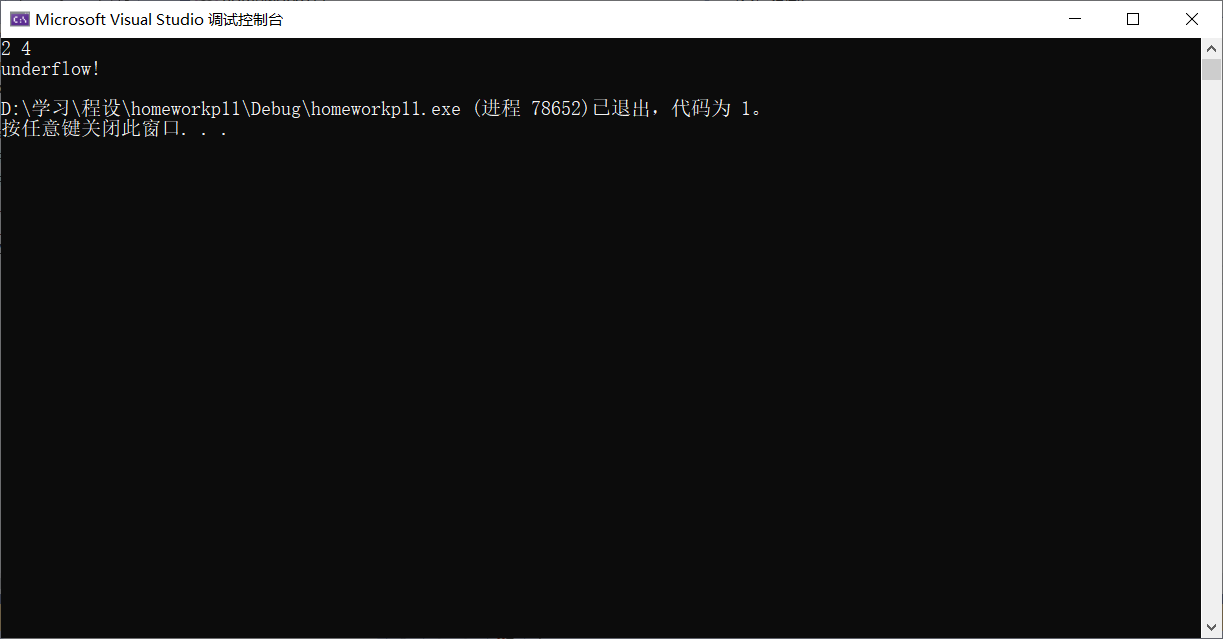
异常1：动态内存申请失败：



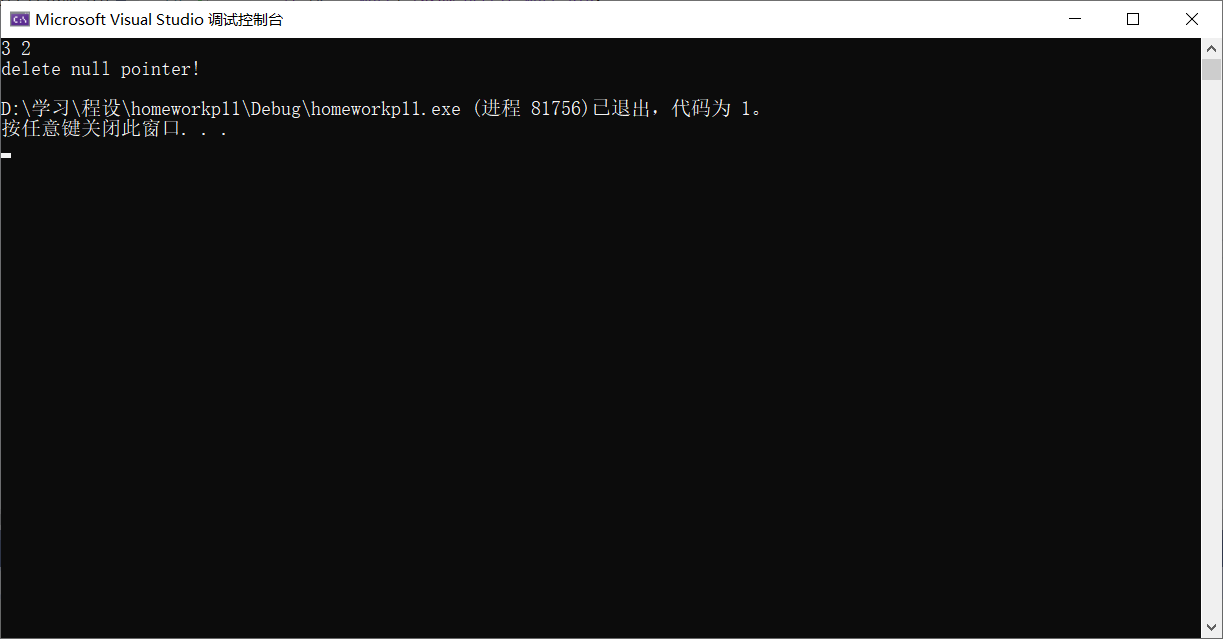
异常2：上溢：



异常3：下溢：

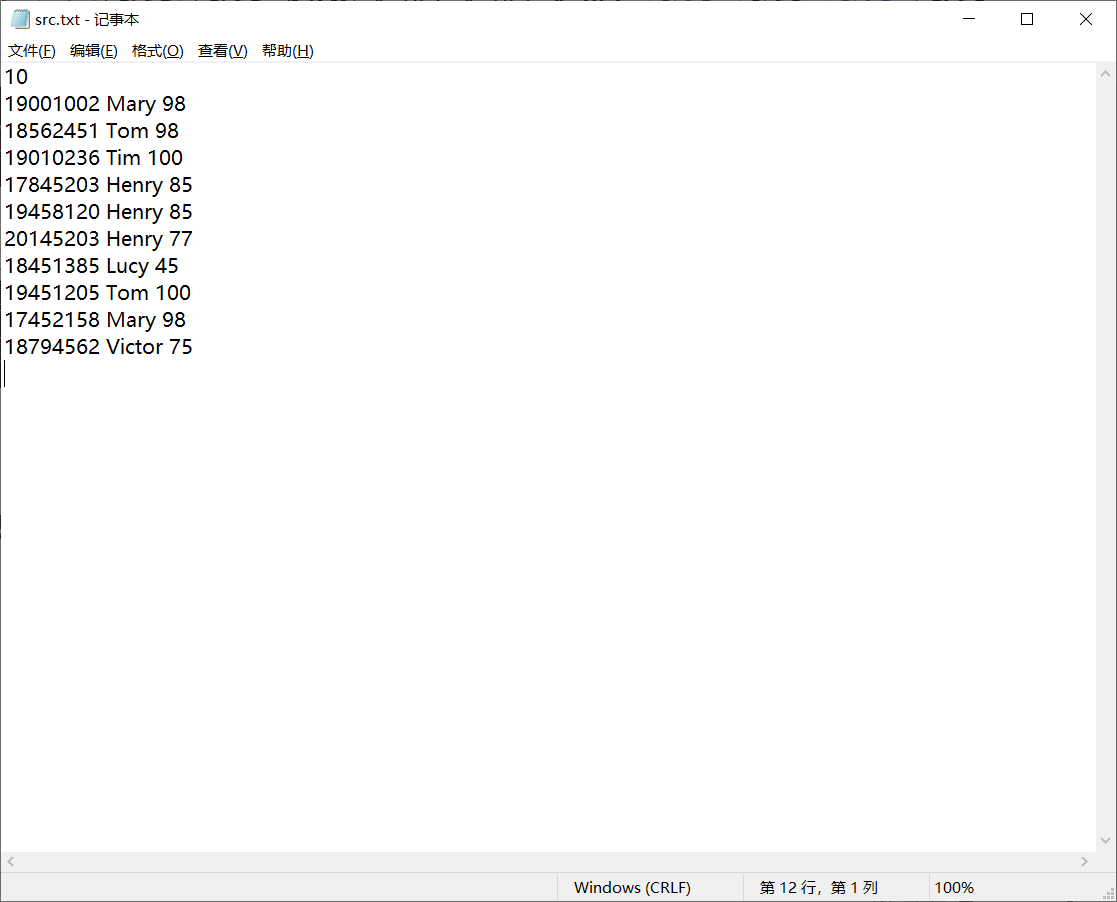


异常4：删除空指针：



2.

学生信息文件：



源代码：

#include <iostream>

#include <fstream>

#include <string>

#include <algorithm>

#include <iomanip>

using namespace std;

struct student

{

int sno;

string name;

int score;

};

bool studcmp(const student& s1, const student& s2)

{

if (s1.score != s2.score) return s1.score > s2.score;

else if (s1.name != s2.name) return s1.name < s2.name;

else return s1.sno < s2.sno;

}

int main()

{

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

ifstream fin(src, ios::in);

if (!fin)

{

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

return 1;

}

int num;

student\* stus;

fin >> num;

stus = new student[num];

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

{

fin >> stus[i].sno >> stus[i].name >> stus[i].score;

}

fin.close();

sort(stus, stus + num, studcmp);

cout << " 学生名单" << endl << "============================" << endl;

cout << setw(10) << "sno|" << setw(10) << "name|" << setw(8) << "score|" << endl;

cout << "----------------------------" << endl;

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

{

cout << setw(9) << stus[i].sno << '|' << setw(9) << stus[i].name << '|' << setw(7) << stus[i].score << '|' << endl;

}

delete[] stus;

return 0;

}

运行结果：

