1.

Code:

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std ;

void main()

{

ifstream is("work.txt", ios\_base::in) ;

int x ;

cout.width(8) ;

for (int i = 1; ; i++)

{

is >> x ;

cout.width(8) ;

cout << x << ' ' ;

if(is.fail()) break ;

if (i % 10 == 0)

{

cout << endl ;

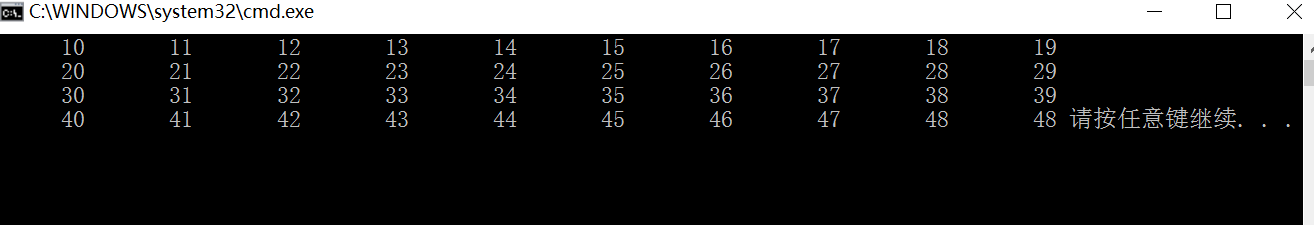
}

}

is.close() ;

}

Result:



2.

Code:

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std ;

static int cnt = 0 ;

void outp(int a)

{

cout << setw(8) << a ;

if((++cnt) % 10 == 0)

{

cout << '\n' << endl ;

}

}

void main()

{

ifstream in("in.txt", ios::in) ;

int a[50] ;

for(int i = 0; i < 50; i++)

{

in >> a[i] ;

}

ofstream out("out.dat", ios::binary) ;

out.write((char\*)a, sizeof(int) \* 50) ;

out.close() ;

ifstream Get("out.dat", ios::binary) ;

Get.seekg(sizeof(int) \* 5) ;

int c, d, skip1, skip2 ;

cout << setiosflags(ios\_base::left) ;

while(1)

{

if(Get.eof())

break ;

Get.read((char\*)&c, sizeof(int)) ;

outp(c) ;

if(Get.eof())

break ;

Get.read((char\*)&d, sizeof(int)) ;

outp(d) ;

if(Get.eof())

break ;

Get.read((char\*)&skip1, sizeof(int)) ;

if(Get.eof())

break ;

Get.read((char\*)&skip2, sizeof(int)) ;

}

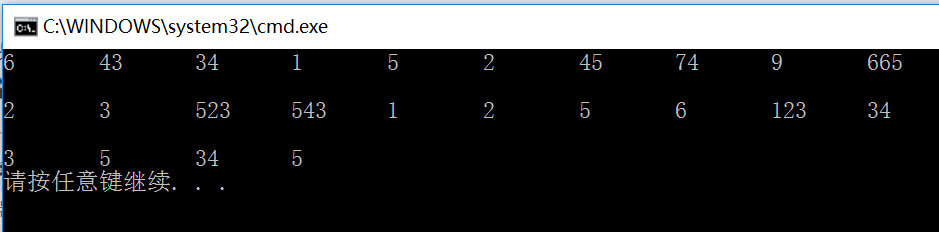
cout << endl ;

Get.close() ;

in.close() ;

}

Result:



3.

Code:

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std ;

class Listnode

{

private:

char\* word ;

Listnode\* next ;

public:

Listnode(char\* s, Listnode\* n = NULL) : word(s), next(n)

{

word = new char[strlen(s) + 1] ; //给word开辟空间

strcpy(word, s) ;

}

~Listnode()

{

delete []word ;

}

friend class wordlist ;

} ;

class wordlist

{

private:

Listnode\* head ;

Listnode\* current ;

public:

wordlist(char\*s)

{

head = new Listnode(s) ;

current = head ;

}

~wordlist()

{

Listnode\* temp ;

temp = head->next ;

delete head ;

while(temp != NULL)

{

current = temp ;

temp = current->next ;

delete current ;

}

}

void insert(char \*s) ;

void Print(ofstream &out) const ;

} ;

void wordlist::insert(char \*s)

{

current = head ;

if(strcmp(head->word, s) >= 0)//插到头部

{

head = new Listnode(s) ;

head->next = current ;//current刚开始停留在原head处

}

else

{

while( current->next != NULL && strcmp(current->next->word, s) < 0)

{

current = current->next ;

}

Listnode \*temp = current->next ;

current->next = new Listnode(s) ;

current->next->next = temp ;

}

}

void wordlist::Print(ofstream &out) const

{

Listnode \* temp = head ;

while(temp != NULL)

{

out << temp->word << endl ;

temp = temp->next ;

}

}

void main()

{

int n ;

cout << "How many strings?" << endl ;

cin >> n ;

ifstream in("in.txt", ios::in) ;

if(!in)

{

cout << "failed" << endl ;

exit(0) ;

}

char s[80] ;

in >> s ;

wordlist w(s) ;

for (int i = 2; i<= n; i++)

{

if(in.eof())

{

cout << "Too much, that's all!" << endl ;

break ;

}

char s[80] ;

in >> s ;

w.insert(s) ;

}

in.close() ;

ofstream out("output.txt", ios::trunc) ;

w.Print(out) ;

out.close() ;

}Result:

