第五章代码

/\*

\* ex5\_1.cpp

\*

\* 1. 编写函数判断两个int数组是否相等，编写程序测试该函数。

\*

\*/

#include <iostream>

using namespace std;

bool arrayCompare(int\* a, size\_t asize, int\* b, size\_t bsize)

{

if(asize != bsize)

return false;

for(size\_t i= 0; i < asize; ++i)

if(a[i]!=b[i])

return false;

return true;

}

int main()

{

int a[] = {1,2,3,4,5,6,7};

int b[] = {3,4,5,6,7,8};

cout << arrayCompare(a,7,b,6) << endl;

cout << arrayCompare(a,6,b,6) << endl;

cout << arrayCompare(a+2,5,b,5) << endl;

cout << arrayCompare(a+2,5,b+1,5) << endl;

}

template <class T1,class T2>

void myfunc(T1 x,T2 y)

{

cout<<x<<" "<<y<<endl;

}

template <class\_T>//通用类型\_T

void bubble(\_T\*items,int count)

{

register int a,b;//大大提高速度

\_T t;

for(a=1;a<count;a++)

for(b=count-1;b>=a;b--)

{

if(items[b-1]>items[b])

t=items[b-1];

items[b-1]=items[b];

items[b]=t;

}

}

/\*

\* ex5\_2.cpp

\*

\* 2. 编写函数，计算一个整数的各位数字之和。用递归和非递归两种方法实现。

\* 函数原型为：int sumDigits(long n);

\*/

#include<iostream>

using namespace std;

int sumDigits(long n){

int sum = 0;

while(n > 0){

sum += n % 10;

n /= 10;

}

return sum;

}

//reursive function

int sumDigitsR(long n){

if (n > 0)

return (n % 10) + sumDigitsR(n / 10);

else

return 0;

}

int main()

{

long n = 123456780L;

cout << n << " digits sum = " << sumDigits(n) << endl;

cout << n << " digits sum = " << sumDigitsR(n) << endl;

return 0;

}

/\*

\* ex5\_7.cpp

\*

\* 7. 编写函数，将两个有序vector合并成一个新的有序vector，函数原型为：

\* vector<int> merge(vector<int> list1, vector<int> list2);

\*

\*/

#include<vector>

#include<iostream>

using namespace std;

vector<int> merge(vector<int> list1, vector<int> list2){

vector<int> vm;

auto it1 = list1.begin(), it2 = list2.begin();

while(it1 != list1.end() && it2 != list2.end()){

if(\*it1 < \*it2)

vm.push\_back(\*it1++);

else

vm.push\_back(\*it2++);

}

while(it1 != list1.end())

vm.push\_back(\*it1++);

while(it2 != list2.end())

vm.push\_back(\*it2++);

return vm;

}

int main()

{

vector<int> v1 = {2, 3, 4, 6};

vector<int> v2 = {1, 3, 5, 7, 9, 13};

vector<int> v3 = merge(v1, v2);

cout << "\nv1: ";

for(auto e : v1) cout << e << " ";

cout << "\nv2: ";

for(auto e : v2) cout << e << " ";

cout << "\nmerge(v1, v2): ";

for(auto e : v3) cout << e << " ";

return 0;

}

/\*

\* ex5\_11.cpp

\*

\* 11. 编写函数，将整数转换成字符串。例如：itoa(-123)，转换结果为："-123"。

\* 函数原型为：string itoa(int);

\*

\*/

#include <iostream>

#include <string>

#include <sstream>

#include <vector>

using namespace std;

string itoa2(int num)

{

ostringstream os;

os<<num;

return os.str();

}

string itoa(int num)

{

vector<char> v;

char sign = 0;

if(num == 0)

return "0";

if(num<0) {

sign = '-';

num = -num;

}

while(num){

v.push\_back(num%10 + '0');

num = num/10;

}

if(sign)

v.push\_back(sign);

string number="";

for(vector<char>::reverse\_iterator ri= v.rbegin(); ri!=v.rend();ri++)

number += \*ri;

return number;

}

int main()

{

cout<<"Please input a number:"<<endl;

int num;

cin>>num;

cout<<itoa(num)<<endl;

cout<<itoa2(num)<<endl;

}